



UNIVERSIDAD NACIONAL DE COLOMBIA

# **Restauración de imágenes mediante un modelo matemático basado en las técnicas de detección de bordes y propagación de texturas**

Nelson Bahamón Cortés

Universidad Nacional de Colombia  
Facultad de Ciencias, Departamento de Matemáticas  
Bogotá, Colombia

2011

# **Restauración de imágenes mediante un modelo matemático basado en las técnicas de detección de bordes y propagación de texturas**

**Nelson Bahamón Cortés**

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título  
de:

**Magister en Ciencias – Matemática Aplicada**

Director: Dr. rer. nat. Hernán Estrada Bustos (QEPD)

Director Ad hoc: Dr. Jorge Mauricio Ruíz Vera

Línea de investigación:

Matemática Aplicada

Universidad Nacional de Colombia

Facultad de Ciencias, Departamento de Matemáticas

Bogotá, Colombia

2011

***Dedicatoria:***

***A mi profesor Hernán Estrada Bustos (Q. E. P. D.),  
Siempre oraré por usted.***

***A mi esposa, Alexandra  
A mis hijas, Juana y Catalina  
Las amo. Que Dios las bendiga***

***A mis Padres Nelson y Amanda. Gracias***

## **Agradecimientos.**

Agradezco en primera instancia a Dios quien me ha dado el don de la vida y me ha permitido culminar este trabajo pues “El querer y el obrar vienen de Dios”.

Agradezco a mi primer director de tesis, el profesor Hernán Estrada Bustos (Q. E. P. D.), quien con su paciencia y constancia me orientó y ayudó en el desarrollo de este trabajo. Falleció cuando el trabajo estaba a punto de terminarse. Me dirigió también en mi trabajo de grado como Físico. Tuve la oportunidad de compartir diversos momentos con él y fue siempre para mí un apoyo y un amigo. Fue una bendición poderlo conocer y aprender de él. He hecho mi mejor esfuerzo para que este trabajo sea no solo de calidad, sino también un homenaje a su memoria.

Agradezco a mi nuevo director de tesis, el profesor Jorge Mauricio Ruíz Vera, también por apoyarme y ayudarme con su dirección a terminar este trabajo.

Agradezco a mi esposa Alexandra García por su amor y confianza. Ella me ha dado alientos para ser perseverante y terminar esta maestría.

Agradezco a mis padres Nelson Bahamón y Amanda Cortés por su amor y ayuda.

## Resumen

Se generó e implementó un modelo matemático para restaurar imágenes basado, en las técnicas actuales de detección de bordes y propagación de texturas. Se utilizó en primera instancia la técnica de síntesis de texturas basada en parches ("Patch Based Texture Synthesis"). Se observó que para lograr la mejor restauración posible se debe lograr un equilibrio entre la propagación del color (textura) y de bordes (estructura) ya que el dar prioridad a una u otra genera resultados muy distintos. En otras palabras, estos dos factores "compiten" entre sí. Así mismo se observó mediante experimentos sencillos que el resultado de la restauración en la imagen depende fuertemente del orden en que se restauren los píxeles de la zona objetivo. Se planteó una función de probabilidad que soluciona ambos problemas, es decir, genera un equilibrio entre la propagación de textura y de estructura y determina el orden de restauración de la zona objetivo. Se presentan 31 (treinta y uno) imágenes con resultados visualmente muy satisfactorios.

**Palabras clave:** Restauración, Textura, Bordes, Imagen, Procesamiento, Modelo, Digital.

## **Abstract**

Was generated and implemented an image inpainting mathematical model based on actual edge detection and texture synthesis techniques. First was used the Patch Based Texture Synthesis technique. There was observed that for obtain a better possible restoration is needed to obtain an equilibrium between color propagation (texture) and edge propagation (structure) because, to give priority one over other generates very different results. In other words, these two factors “rival” between them. Therefore was observed from simple experiments that image inpainting depends strongly from order restoration pixels into objective zone. Was proposed a probability function that solve booth problems, this means, to generate equilibrium between texture and structure propagation and, to determine the order restoration objective zone. Here are presented 31 (thirty one) images with very satisfactory visual results.

**Keywords:** Inpainting, Restoration, Texture, Edge, Image, Processing, Digital.

# Contenido

<i>Dedicatoria:</i> .....	3
Agradecimientos.....	4
<b>1. Introducción</b> .....	8
<b>2. Elementos en procesamiento de imágenes (Marco Teórico).</b> .....	9
2.1. Concepto de restauración de imágenes.....	10
2.2. Algunos aspectos sobre imágenes digitales. ....	15
2.3. Conceptos básicos e investigación en procesamiento digital de imágenes. ....	18
2.4. Uso de ecuaciones diferenciales en el procesamiento de imágenes. ....	23
2.5. Texturas en el procesamiento digital de imágenes. ....	31
2.6. Detección de bordes. ....	37
<b>3. Modelo matemático propuesto</b> .....	43
3.1. Estructura matemática del modelo. ....	43
3.2. Imágenes restauradas mediante el modelo propuesto. ....	64
<b>4. Conclusiones</b> .....	118
<b>5. Bibliografía</b> .....	119
Anexo 1 (Parámetros) .....	129

# 1. Introducción.

Uno de los campos de la modelación matemática en los que actualmente se está publicando y haciendo investigación es en el del tratamiento y procesamiento digital de imágenes. [47][19].

Dentro de este campo, un área de particular interés, es el de la restauración de imágenes. Dicho interés se ha incrementado debido a la posibilidad de abordar el problema desde diferentes conceptos propios del procesamiento digital de imágenes, tales como la detección de bordes, la propagación de texturas y la utilización de ecuaciones diferenciales en algunos problemas particulares.

La restauración de imágenes es sin duda un campo de especial interés tanto desde el punto de vista matemático como comercial. En las industrias del cine y la publicidad es frecuente la necesidad de restaurar imágenes que por el paso del tiempo y/o falta de cuidado, se han deteriorado. Dichas imágenes pueden tener un valor histórico. Algo similar pasa en el campo del arte, en el cual es frecuente la necesidad de restaurar determinadas obras. En este sentido existe una discusión abierta entre quienes consideran que una obra de valor artístico se debe restaurar y quienes consideran que solo se debe conservar.

En la sección 2.1 se profundiza un poco más en el concepto de restauración de imágenes y se muestran algunos ejemplos que ilustran como la restauración se puede hacer de forma manual e intuitiva o mediante un modelo matemático, como se quiere hacer en este trabajo.

De esta forma los objetivos que se persiguen en este trabajo son los siguientes:

*Objetivo general (problema):*

Generar y poner en práctica un modelo matemático basado en las técnicas actuales de procesamiento digital de imágenes para detección de bordes y propagación de texturas, que permita restaurar imágenes.



### *Objetivos específicos.*

1. Generar y solucionar numéricamente un algoritmo que permita resolver el problema planteado.
2. Mediante el algoritmo propuesto hacer pruebas de restauración en imágenes sencillas bien conocidas.
3. Probar el algoritmo con imágenes complejas como fotografías.

De esta manera la estructura de este documento es la siguiente: En primera instancia (capítulo 2) se explica el marco teórico que sirvió como base para el desarrollo del modelo matemático propuesto en este trabajo y que constituye al mismo tiempo una visión global del estado del arte en este campo. En segunda instancia (capítulo 3) se explica y muestran los resultados del modelo matemático. Finalmente (capítulo 4) se concluye.

## **2. Elementos en procesamiento de imágenes (Marco Teórico).**

Se explican en este capítulo, diferentes elementos previos que sirven como marco teórico del problema que se aborda en este trabajo. En primera instancia y a manera de información se abordan algunos aspectos sobre imágenes digitales (sección 2.2) y su relación con el procesamiento digital de imágenes como campo de investigación y de producción activa de publicaciones (sección 2.3). Se explica también el concepto de restauración de imágenes (sección 2.1) que es el objetivo de este trabajo. En segunda instancia se muestran elementos que fueron fundamentales para proponer la estructura matemática del modelo desarrollado. Se explica el uso de ecuaciones diferenciales en procesamiento digital de imágenes (sección 2.4) y luego elementos muy importantes como son la síntesis de texturas (sección 2.5) y detección de bordes (sección 2.6).

Vale la pena recalcar estos últimos tres elementos:

En primer lugar, lo que respecta a el creciente uso y aplicación de ecuaciones diferenciales en el procesamiento digital de imágenes (sección 2.4). En esta parte se hace énfasis en la ecuación de difusión. Aunque al momento de proponer el modelo matemático de este trabajo, no se utilizó dicha ecuación, se consideró pertinente colocar esta sección, por la importancia que hay en la relación de dicha ecuación con el filtrado gaussiano y sus aplicaciones. En sí, se puede decir que todos los trabajos en procesamiento digital de imágenes utilizan ecuaciones diferenciales, aunque lo hagan en forma implícita, ya que siempre aparece el concepto de razón de cambio de un punto (pixel) a otro.

En segundo lugar, el uso de texturas en procesamiento digital de imágenes (sección 2.5). En esta parte es importante el concepto de texel, ya que la restauración no se hizo pixel a pixel sino por pequeñas regiones, que en si, se asemejan al concepto de texel pero en el trabajo se les llamaron parches por no tratarse exactamente de lo mismo. En esta parte es muy importante el concepto de síntesis de texturas y concretamente una técnica especial basada en parches ("*Patch based texture synthesis*") ya que esta forma parte importante del modelo que se propone en este trabajo.

En tercer lugar se explican algunos elementos relacionados con detección de bordes (sección 2.6). A este respecto vale la pena resaltar entre otras, las técnicas relacionadas con cruces por cero ("*Zero crossing*") y de curvas de nivel ("*Level set*"). De cualquier manera en estas técnicas es fundamental el concepto de gradiente, que se usará en este trabajo.

## **2.1. Concepto de restauración de imágenes.**

Como su nombre lo indica, la restauración es el proceso de reconstruir partes perdidas o deterioradas de imágenes o videos. Para el caso de las imágenes esta labor es ya conocida de tiempo atrás desde el punto de vista técnico, pues puede ser realizada en forma intuitiva por un artista experto. Sin embargo en el mundo de las imágenes y

videos digitales este concepto ha cambiado pues se han generado algoritmos matemáticos de cierta complejidad que permiten hacer lo mismo mediante un software. [6]

Una manera de encontrar información sobre restauración de imágenes es mediante la palabra “inpainting” que traduce algo así como “pintar dentro” y es el término usado en inglés para referenciar este proceso que actualmente constituye un campo muy activo de investigación. [6][32][35][76]

Claramente la restauración de imágenes permite solucionar diferentes problemas particulares como eliminar ruido, mejorar aspectos como el brillo, el color y los detalles de la imagen. Las posibilidades de restaurar una imagen abren también las puertas a la posibilidad de restaurar videos por lo cual es un campo de importancia en la industria del cine, especialmente cuando se trata de películas deterioradas.[80]

En fotografía es común usar estas técnicas para quitar el efecto de ojos rojos, quitar textos impresos como por ejemplo la fecha o incluso para quitar objetos que no se desea que aparezcan en la foto.

Las técnicas de restauración se utilizan también para recuperar información perdida, en la transmisión de imágenes o videos.

En un primer desarrollo básico relacionado con la restauración, se ha utilizado la aplicación de filtros para la eliminación de ruidos.

En algunos casos, los métodos para restaurar imágenes, se han enfocado en el uso de ecuaciones diferenciales.

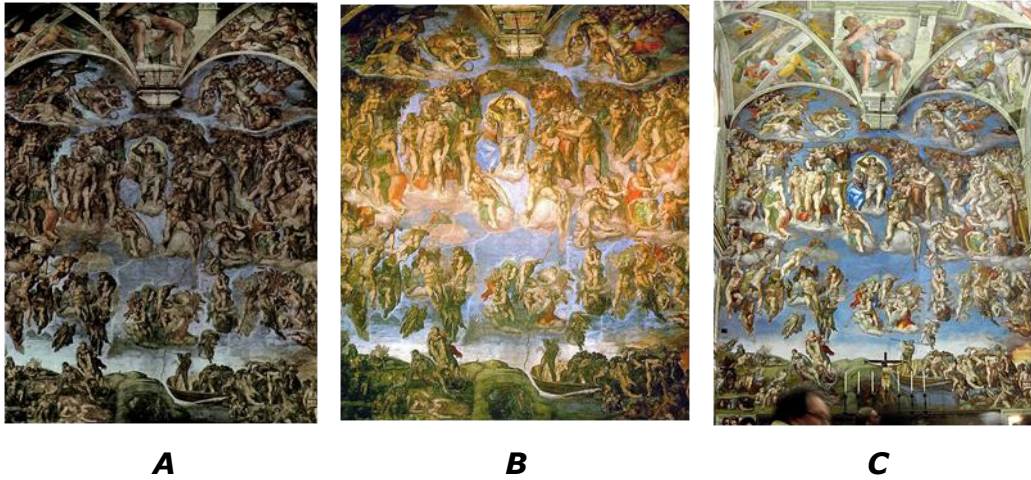
A continuación se muestran ejemplos diferentes de imágenes restauradas. La *Imagen 2.1-1* corresponde a la restauración de una obra de arte (El Juicio Final en la Capilla Sixtina). En este caso se trata de una labor hecha a mano por expertos [34]. Es interesante el hecho de que muchos pensaban que la obra original de Miguel Ángel era tan oscura como se observa en la *Imagen 2.1-1.A*. Sin embargo al momento de restaurarla se observa que se trata de una obra luminosa, llena de colores vivos y claros. En el campo del arte, existe una extensa discusión con respecto al tema de la

restauración [60][73]. Está la contraposición entre restauración y conservación, es decir la duda de hasta qué punto una obra de arte se debe restaurar o simplemente proteger y conservar. Igualmente se discute en caso de intervenir la obra, con qué criterios se debe hacer, es decir si prima el punto de vista del historiador, el científico o el crítico de arte. [70][21]. A este respecto el presente trabajo es interesante, porque al realizar la restauración de una imagen digital de la obra de arte en cuestión, mediante un modelo matemático, se hace a un lado el asunto de la subjetividad de quienes intervengan en el proceso.

La *Imagen 2.1-2* corresponde a una corrección del efecto de ojos rojos. Este efecto se da cuando el sitio donde se toma la foto está poco iluminado y la pupila de la persona se encuentra dilatada. En la imagen mostrada, la corrección se hace mediante un software comercial especializado en fotografía [28].

La *Imagen 2.1-3* corresponde a la restauración de una fotografía mediante la implementación complementaria de un modelo matemático en un programa de cómputo y técnicas manuales de restauración en fotografía. [48].

El modelo propuesto en este trabajo permite abordar imágenes como las del segundo y tercer caso, es decir, imágenes que sean digitales o que en su defecto, puedan ser digitalizadas.



**Imagen 2.1-1.** *El Juicio final (Capilla Sixtina) de Miguel Ángel. Antes (A), durante (B) y después (C) de ser restaurada. En este caso la restauración se hizo por expertos, mediante la aplicación de diferentes químicos. Tomado de [66]*



**Imagen 2.1-2.** *Corrección del efecto ojos rojos mediante un software. A antes de la corrección. B después de la corrección. Tomado de [28]*



**Imagen 2.1-3.** Restauración de una fotografía mediante un modelo matemático y técnicas manuales de restauración en fotografía. **A**, antes de la restauración. **B**, después de la restauración. Tomado de [48]

En el campo de la restauración manual y artística existen diferentes técnicas de restauración, las cuales dependen fundamentalmente del cuadro y del restaurador [48]. Sin embargo se pueden definir los siguientes criterios o pasos simples en forma más o menos general:

- La zona que se va a restaurar se rellena utilizando la información que brinda el resto de la imagen.
- La estructura de la frontera a la región a restaurar se continúa, prolongando las líneas o bordes.
- Una vez que se han generado regiones en el interior de la región a restaurar se rellenan con el color correspondiente de la frontera.
- Para finalizar se pintan los detalles necesarios para que el cuadro se vea uniforme y no se note la restauración

Bien. Así en el campo de las imágenes digitales lo que se busca es traducir las técnicas manuales de los restauradores profesionales de cuadros, a una versión digital de

las mismas. La idea de hacer este trabajo en forma digital, es lograr que la zona a restaurar de la imagen quede completamente fusionada a la zona conocida de la imagen de forma que al observador no le sea posible detectar en un visor común, la restauración. [6][15]

En general, el resultado siempre dependerá de la información brindada por las zonas conocidas de la imagen (no solo de la frontera), por esto el problema de la restauración se conoce en algunos ámbitos como un tipo de interpolación. [25][24]

Las técnicas de restauración de imágenes digitales han evolucionado con el tiempo. Inicialmente se enfocaron hacia técnicas en el dominio de la frecuencia, utilizando la transformada de Fourier. Recientemente los trabajos se han basado en métodos algebraicos que involucran el manejo de grandes sistemas de ecuaciones. [65]

De acuerdo con todo esto, es claro que el problema de restaurar imágenes mediante un modelo matemático es un campo de investigación donde se está publicando actualmente. El problema es de interés tanto desde el punto de vista matemático, por el potencial que tienen los conceptos matemáticos previos, propios del procesamiento digital de imágenes, como desde el punto de vista industrial o comercial, por la necesidad de desarrollar algoritmos que permitan restaurar imágenes deterioradas o alteradas y convertirlas en imágenes de buena calidad.

## **2.2. Algunos aspectos sobre imágenes digitales.**

En sí, el interés que hay presente en los diferentes trabajos en torno al procesamiento digital de imágenes es el de pasar de una información de tipo local, que está contenida en el nivel de intensidad de un punto de la imagen, a una información global que puede brindar la imagen según la aplicación que se esté trabajando. Por lo general dicha información global está relacionada con los objetos contenidos en la imagen.

Para el caso de una imagen digital los puntos en cuestión son los pixeles que conforman la imagen. Sin importar el tipo de proceso que se esté haciendo, se presupone

por lo general que la imagen es lo suficientemente grande para considerar que la información que brinda un pixel es de carácter infinitesimal. Esta suposición es bastante razonable si se considera el tamaño creciente de las imágenes digitales en los diferentes dispositivos de procesamiento (cámaras, filmadoras). Se podría decir actualmente que una imagen pequeña es de unos  $400 \times 300 = 120000$  pixeles. Comercialmente, para saber la resolución máxima de una cámara digital se referencia en términos del total de pixeles y no de las dimensiones de la matriz (número de filas y de columnas). Así, una cámara con resolución de 5M (5 Megapixeles) toma fotografías por ejemplo de  $2560 \times 1920 = 5000000$  pixeles. Una buena resolución en una fotografía significa que el número de pixeles es alto y que si la imagen se amplía (por ejemplo para hacer una impresión) sigue viéndose con nitidez.

Así una imagen digital consiste en sí, en un arreglo matricial de puntos o pixeles. Si se tiene una imagen monocromática (por ejemplo en escala de grises) se puede pensar en términos de una sola matriz. Si se tiene una imagen en colores, lo que se tiene es realmente la superposición de tres matrices, cada una con un componente de color diferente. Así por ejemplo en el espacio de colores RGB, cada imagen se compone por la superposición de tres matrices o imágenes de cada una de las componentes de color rojo (R = red), verde (G = green) y azul (B = blue).

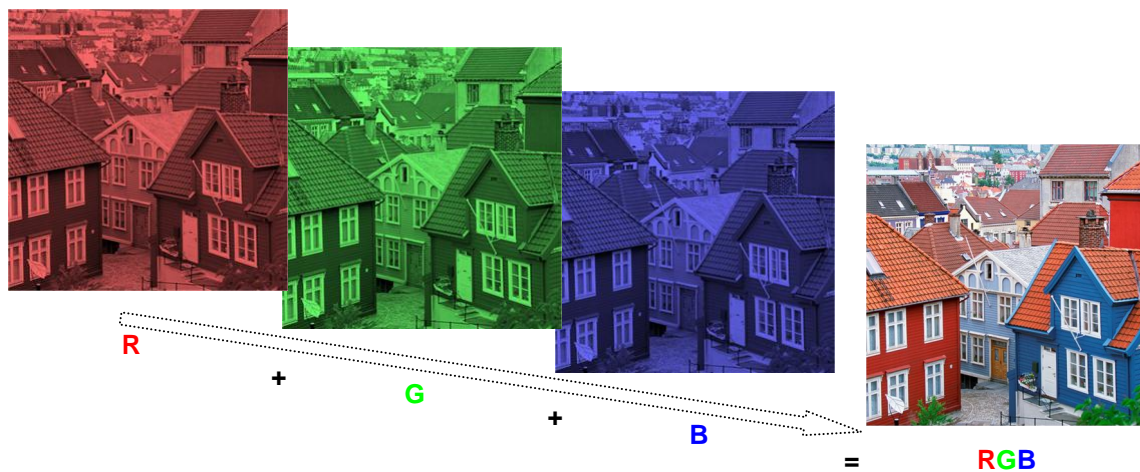
La *Figura 2.2-1*, ilustra la composición de una imagen por pixeles. La imagen es una función bidimensional  $B(x,y)$  dada por sus coordenadas espaciales y su brillo. En forma discreta se tendrá  $B(i,j)$ . Así los índices de las filas y las columnas indican un punto (pixel) específico de la imagen y cada pixel en la matriz tiene un determinado valor de intensidad o brillo  $B(i,j)$ . Por lo general, para hacer referencia al tamaño de la imagen digital se dice que es de  $M \times N$  pixeles, es decir, M denota el número de columnas y N el de filas. Para el caso de la imagen de la *Figura 2.2-1*, su dimensión es de  $440 \times 443$  pixeles. Es decir que tiene 194920 pixeles.

La *Figura 2.2-2* ilustra la superposición de matrices o imágenes en el espacio de color RGB para dar lugar a una imagen digital en color. Como se observa, es la misma imagen de la *Figura 2.2-1*.





**Figura 2.2-1.** Estructura de una imagen digital.



**Figura 2.2-2.** Superposición de las componentes de color en el espacio de color **RGB**.

Así, los instrumentos para la adquisición de imágenes (cámaras, videograbadoras, etc.) convierten la imagen visual de un objeto físico o de una escena, así como sus atributos intrínsecos, en una matriz de datos digitalizados.

## **2.3. Conceptos básicos e investigación en procesamiento digital de imágenes.**

Se puede decir que el desarrollo tecnológico de los computadores ha generado en los últimos tiempos la aparición de nuevas disciplinas científicas como lo es por ejemplo el “Procesamiento digital de imágenes”. [47]

Inicialmente surgió lo que hoy se conoce como teoría de “bajo nivel” [44]. Esta consiste en estudiar una imagen conociendo como es el nivel de intensidad de un punto de la misma con respecto a sus vecinos sin tener conocimiento de otra información de “alto nivel”, es decir, sin tener conocimiento de los objetos presentes en la imagen.

El hecho que permitió desarrollar esta teoría de bajo nivel es que la visión se puede descomponer en diferentes etapas [44], de las cuales, en la primera, se detecta dicha diferencia de intensidad entre puntos en la imagen y sus vecinos. Este es un proceso que se hace en forma automática, en las primeras fracciones de segundo con la información que llega a la retina. Recientemente se han hecho diferentes trabajos donde se utilizan ecuaciones diferenciales para procesar la imagen. Muchos de estos trabajos están enmarcados dentro de la teoría de bajo nivel puesto que lo que se hace es aplicar determinados operadores diferenciales que tejen relaciones entre diferentes puntos de la imagen y sus vecinos sin utilizar información particular y global sobre la imagen. [65]

Vale la pena insistir en esto. Lo que determina perceptualmente la información contenida en una región de la imagen es la diferencia en el nivel de brillo de un pixel con sus vecinos y no la información dada por el pixel mismo. De hecho se podría decir que, en todas las técnicas de procesamiento digital de imágenes aparecen operadores diferenciales que involucran la relación entre pixeles vecinos, aunque dichos operadores no se mencionen explícitamente. Esto lleva a un marco teórico inmediato que es el de las ecuaciones diferenciales. Respecto de este punto se hará una mención especial en la sección 2.4.

Así, en el procesamiento de imágenes de bajo nivel hay muy poco uso del conocimiento que se pueda tener sobre el contenido de las imágenes. Normalmente para el procesamiento de bajo nivel se conocen cuatro pasos:

- Adquisición de la imagen.
- Pre-procesamiento.
- Segmentación de la imagen.
- Descripción y clasificación de objetos.

Por otro lado en el procesamiento de alto nivel existe la capacidad de realizar toma de decisiones respecto del contenido de las imágenes [16]. En este caso se puede pasar de trabajar con puntos (que para el caso de las imágenes digitales son píxeles) a trabajar con contornos que permitan generar curvas, y con regiones [19].

En sí, el procesamiento digital de imágenes consiste en un conjunto de operaciones que se llevan a cabo sistemáticamente sobre la imagen con el propósito de obtener mediciones cuantitativas de la misma. Esta información cuantitativa permite describir la imagen y extraer características que se utilizan para perfeccionar y/o detallar la imagen. [32][57]

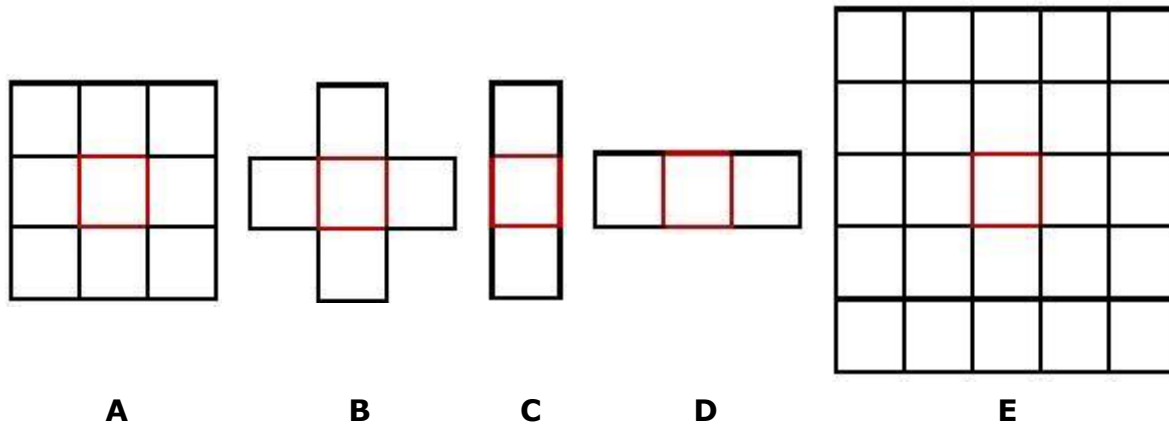
Una característica o atributo de la imagen es algo que permite tomar decisiones basándose en los objetos que hay en la misma. Algunas características son inmediatas y naturales, pues tienen que ver con la apariencia visual de la imagen, pero otras son de tipo artificial ya que surgen como resultado de operaciones que se realizan sobre la imagen. [57][64]

Las operaciones a realizar sobre una imagen determinan un procedimiento definido. Dicho procedimiento está estrechamente ligado con la aplicación particular a que se quiere llegar. El procedimiento para una determinada aplicación puede no servir para otra. Para desarrollar estos procedimientos es conveniente que las características (o atributos) de la imagen estén bien definidos. En particular que los bordes estén bien delimitados y el color y el brillo sean uniformes [36][64]. En las secciones 2.5 y 2.6 se especifican algunos ejemplos de procedimientos y operaciones para realizar síntesis de texturas y detección de bordes respectivamente.

Un concepto que vale la pena mencionar por su importancia en procesamiento digital de imágenes es el de *filtro*. Un filtro es una forma específica de hacer operaciones sobre los píxeles de una imagen para obtener una nueva. A la nueva imagen se le podría llamar imagen filtrada. En sí, estas operaciones hechas sobre cada píxel no dependen solo de su brillo (o nivel de gris) sino también del de sus vecinos en la imagen original. La forma como se escogen los píxeles vecinos para realizar dichas operaciones determina una matriz que se denomina máscara y que se aplica haciendo un barrido sistemático sobre todos los píxeles de la imagen. En la *Figura 2.3-1* se muestran ejemplos de posibles máscaras. Es claro que en sí, el filtro no se determina solamente por la forma de escoger los vecinos, es decir la máscara, para realizar las operaciones, sino por las operaciones mismas. Un filtro muy conocido por sus propiedades y de obligatoria mención en cualquier trabajo relacionado con procesamiento digital de imágenes es el filtro *gaussiano*. [78]

En general después de aplicar un filtro, se obtiene una nueva imagen donde se han suavizado o realzado detalles de la misma, o incluso minimizado efectos de ruido [4][50][30]. El ruido en sí, se define como cualquier aspecto en la imagen, dato o resultado intermedio, que no es interesante para el cómputo que se quiere llevar a cabo y para la información que se pretende obtener.

El filtrado de una imagen se puede hacer no solo en el dominio del espacio sino también en el dominio de la frecuencia, tras aplicar la transformada de Fourier a la imagen. Este aspecto se retomará en el sección 2.6.



**Figura 2.3-1.** Algunas máscaras posibles.

A continuación se especifican algunos de los filtros más utilizados en la literatura y se coloca una breve descripción [46]:

**Filtro gaussiano:** Por sus propiedades matemáticas se puede decir que es el filtro más útil y por eso se explica primero en cualquier libro de procesamiento digital de imágenes. Estas propiedades y el filtro mismo se explicarán con más detalle en la sección 2.4. Este filtro es práctico para remover ruido pero tiene el inconveniente de que empaña la imagen y hace perder los detalles más finos de la misma.

**Filtro mediana:** Consiste en reemplazar el pixel del centro, por la mediana del brillo de los pixeles de la máscara. Este filtro reduce el empañamiento de los bordes. Tiene la ventaja de que no es afectado significativamente por pixeles individuales ruidosos y que la eliminación de ruido es bastante buena. Tiene los inconvenientes de que su efectividad disminuye cuando el número de pixeles con ruido es superior a la mitad de los pixeles de la máscara y de que es más lento que otros filtros.

**Filtro de suavizado direccional (preservación de bordes):** Al utilizar filtros de suavizado (como el gaussiano) para eliminar ruido se produce también un efecto de empañamiento de los bordes. Este filtro tiene el propósito de eliminar este efecto.

**Filtro de suavizado conservador:** Este filtro tiene un bajo poder de eliminación de ruido a cambio de preservar mejor los detalles de la imagen. El filtro es muy simple. Si el pixel central de la máscara tiene el mayor brillo, se reemplaza su valor por el que le sigue de sus vecinos. Algo similar se hace si tiene el menor brillo. Si esto no se da, se deja como está.

**Realce de contraste:** Como su nombre lo indica, este filtro tiene como objeto realzar el contraste de la imagen. El objetivo es mejorar la discriminación visual entre dos objetos. El contraste entre los dos objetos se puede definir como la razón entre sus niveles medios de brillo. El proceso se hace mediante una transformación que conserva las proporcionalidades (radiométrica) basada en el histograma de la imagen.

**Filtros pasa bajo y pasa alto:** Estos filtros se realizan realmente en el espacio de frecuencias por lo cual requieren que primero se aplique la transformada de Fourier. Como su nombre lo indica, lo que hacen es quitar altas y bajas frecuencias respectivamente. Los cambios suaves de tonalidad de la imagen están asociados con las frecuencias bajas mientras que los cambios bruscos (bordes) con frecuencias altas. Esto implica que un filtro pasa alto pueda ser usado en la detección de bordes. En la sección 2.6 se retoma esta idea.

**Filtro Susan:** Este filtro utiliza varias de las técnicas de eliminación de ruido existentes en la literatura pero al mismo tiempo logra preservar los bordes, por lo cual los resultados son bastante satisfactorios. Utiliza un núcleo circular de pixeles para el cálculo del nuevo valor en el pixel central de la máscara. De ahí deriva su nombre (Smallest Univalve Segment Assimilating Nucleus). Para mayor información ver [46].

En el procesamiento digital de imágenes se tienen diferentes aplicaciones y problemas entre los que se pueden mencionar:

- Representación.
- Transformación.
- Modelado.

- Restauración.
- Reconstrucción.
- Análisis.
- Compresión de datos.

Este trabajo cae en el campo de la restauración de imágenes.

## **2.4. Uso de ecuaciones diferenciales en el procesamiento de imágenes.**

Se considera pertinente dedicar esta breve sección para resaltar el uso de ecuaciones diferenciales en el procesamiento digital de imágenes. Especialmente porque prácticamente en todas las aplicaciones y procedimientos las ecuaciones diferenciales están presentes aunque sea de una manera implícita. Por otro lado se hará referencia también al filtro gaussiano mencionado anteriormente, así como su importancia y su interesante relación con la ecuación de difusión. [65]

Vale la pena decir que el procesamiento digital de imágenes utiliza un amplio espectro de técnicas de las matemáticas y especialmente de la matemática aplicada. Las áreas que se utilizan son fundamentalmente: álgebra lineal, cálculo diferencial e integral en una y varias dimensiones, ecuaciones diferenciales ordinarias y parciales, ecuaciones integrales (transformadas integrales), probabilidad y estadística [5][16]. Esta sección se centra en las ecuaciones diferenciales.

Como ya se dijo, un problema fundamental en el trabajo en procesamiento digital de imágenes es la necesidad de pasar de una información de carácter totalmente local, como el nivel de intensidad de un pixel, a una información de carácter global como puede ser la descripción de los objetos presentes en la imagen [47]. Esto equivale a pasar del procesamiento de imágenes en la teoría de bajo nivel al procesamiento de imágenes en la teoría de alto nivel [16].

Lo que ha permitido el desarrollo de esta teoría de bajo nivel es el hecho de que en primera instancia el reconocimiento de una imagen (durante la primera fracción de segundo) se hace mediante la detección de cambios locales de intensidad sin que se utilice ninguna información sobre los objetos como tal, presentes en la imagen. [5]

En este orden de ideas, la información contenida en un pixel tiene muy poca importancia en comparación con la información que brindan las relaciones de vecindad con los pixeles próximos. Esto es lo que determina perceptualmente la información contenida en una región de la imagen. Esto implica que de forma implícita o explícita, todos o casi todos, los trabajos que tienen que ver con tratamiento de imágenes utilizan conceptos como el de gradiente y laplaciano. Si se quiere formalizar esto matemáticamente hablando, se cae inmediatamente en el campo de las ecuaciones diferenciales [47].

Así, mediante un modelo matemático concretado en una o varias ecuaciones diferenciales se proponen unas reglas que determinan el modo de interactuar entre sí de los valores locales de la imagen para generar una información más global y compacta de la misma. Dicho en forma simple, las ecuaciones diferenciales permiten, dar el salto de lo local a lo global, o de una teoría del tratamiento digital de imágenes de bajo nivel a una teoría de alto nivel. Y lo más importante, permite hacerlo, no dentro de un marco intuitivo sino dentro de un modelo matemático natural. [4]

Tal vez el primer ejemplo de una formalización de este tipo es el filtrado gaussiano. El filtro gaussiano es un filtro, que implementa máscaras (ver sección 2.3) donde la operación aplicada es la siguiente gaussiana:

$$G(x,y) = e^{-\frac{(x+y)^2}{2\sigma^2}} \quad (2.4.1)$$

Donde  $x$  y  $y$  son las coordenadas de la imagen y  $\sigma$  la desviación estándar de la probabilidad de distribución asociada. Dicha desviación es el único parámetro del filtro y puede estar relacionado con el tamaño de la máscara, es decir con el área determinada por los pixeles vecinos para aplicar el filtro. Los pixeles más lejanos al pixel central tienen



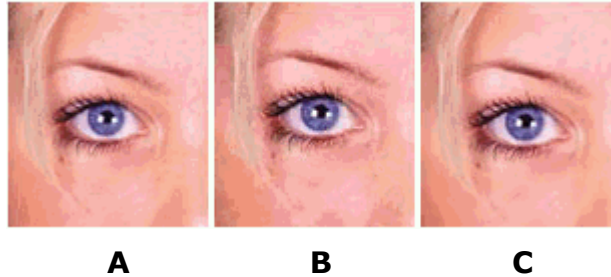
un menor peso en la operación. Desde el punto de vista estadístico se puede decir que los pixeles mas allá de tres valores de  $\sigma$  tienen una influencia despreciable. [4]

Este filtro ha tenido una singular acogida en la comunidad de trabajo sobre procesamiento digital de imágenes por ciertas propiedades que lo hacen especial. Se explican a continuación:

Como ya se ha mencionado, en la teoría de bajo nivel (debida principalmente a David Marr [44]) lo que importa son las relaciones de vecindad entre pixeles y no la información dada por el propio pixel.

De esta manera, lo primero que hay que hacer al comenzar a procesar una imagen, es detectar los cambios de intensidad. A este respecto aparecen entonces dos elementos importantes [47][11]:

- Los cambios de intensidad ocurren en diferentes escalas, por lo tanto para detectarlos en forma óptima y satisfactoria es necesario emplear operadores (máscaras) de diferentes tamaños. Esto es lo que da lugar a lo que se conoce como filtrado o análisis multiescala (en inglés Scale-Space). Existen diversas aplicaciones de análisis multiescala utilizando la transformada wavelet, especialmente con el propósito de comprimir imágenes [40]. Se muestra un ejemplo en la siguiente imagen.



**Imagen 2.4-1.** Ejemplo de análisis multiresolución. En este caso se utiliza la transformada wavelet en el análisis. Una de las principales aplicaciones de esta transformada en imágenes es la compresión de las mismas con poca pérdida de la calidad. Se tiene: **A:** Imagen original. **B:** Imagen comprimida (relación 200:1) con tecnología JPEG estándar. **C:** Imagen comprimida (misma relación) con tecnología JPEG – 2000 que utiliza la transformada wavelet. Imagen tomada de [40]

- Un cambio brusco en la intensidad, dará lugar a un máximo o un mínimo en la primera derivada de la función de intensidad o brillo  $B(x,y)$  o equivalentemente un cero en la segunda derivada. Estos ceros (en inglés “zero-crossing”), son importantes en muchos análisis por ejemplo para la detección de bordes. Un algoritmo conocido para detección de bordes basado en cruces por cero es el algoritmo de Canny. En la *Imagen 2.6-4* (sección 2.6) se muestran algunos resultados utilizando este algoritmo.

Así, para detectar adecuadamente los cambios de intensidad, lo deseable es tener un filtro con las siguientes dos características:

- Debe ser un operador diferencial que extraiga la primera y la segunda derivada espacial de la imagen.
- Debe poderse ajustar fácilmente para ser aplicado a diferentes escalas. De esta manera se podrían aplicar filtros grandes para detectar bordes de sombra más bien difuminados y filtros más pequeños para detectar bordes finos que estén bien definidos en la imagen.

El filtro por excelencia que cumple con estos requisitos es el gaussiano. Esto ha generado que la convolución con funciones gaussianas sea una de las herramientas más utilizadas al momento de analizar una imagen, en la comunidad científica internacional de visión artificial.

Concretamente el filtro que propuso Marr fue el laplaciano de una función gaussiana de desviación típica variable. Esto lo llevó a sugerir que la representación de los ceros de la segunda derivada obtenidos a todas las escalas posibles son como una huella digital que caracteriza la imagen y que esta puede reconstruirse a partir de la información suministrada por dichos ceros.

Vale la pena mencionar que se puede formalizar matemáticamente el análisis multiescala de la siguiente manera: Una imagen no es más que una función:

$$B: \mathbb{R}^n \rightarrow \mathbb{R} \quad (2.4.2)$$

Donde  $n$  es por lo general 2 y se tendrá  $B(x, y)$ . Así el análisis multiescala es una familia de transformaciones

$$\{T_t\}_{t \geq 0} \quad (2.4.3)$$

Donde el parámetro  $t$  representa la escala. Si  $t$  es 1 la imagen resultante después de aplicar el filtro tendrá el mismo tamaño y resolución que la imagen original. Si  $t$  es mayor que 1 se tendrá que para una imagen  $B$ ,  $T_t(B)$  es una nueva imagen que representa a  $B$  pero a una escala más gruesa determinada por  $t$ .

Otro elemento importante del filtrado gaussiano tiene que ver con el análisis de Fourier que en determinadas técnicas se utiliza sobre imágenes [30]. En el análisis de Fourier (téngase en cuenta el principio de incertidumbre), se tiene que las funciones mejor localizadas en tiempo y frecuencia son las gaussianas.

Bien, hasta el momento se ha hecho un pequeño esbozo de las ventajas del filtro gaussiano. Por otro lado a mediados del siglo XIX [5] se descubrió que algunos métodos de resolución de ecuaciones diferenciales ordinarias, correspondían a casos especiales

de un método general basado en la invarianza de una ecuación diferencial bajo la acción de un grupo de transformaciones. Las técnicas desarrolladas en la teoría de grupos de Lie permiten determinar cuál es el grupo de transformaciones invariantes que satisface una ecuación diferencial.

Se quiere mostrar ahora una relación muy interesante entre la ecuación de calor y su respectivo grupo de transformaciones invariantes con el filtrado gaussiano.

Vale la pena recordar la ecuación de calor, la cual es una ecuación diferencial en derivadas parciales. Es la siguiente:

$$\frac{\partial u}{\partial t} - \alpha \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) = 0 \quad (2.4.4)$$

Donde:  $u$  = Calor o variación de la temperatura.  
 $\alpha$  = Difusividad térmica. Depende del material.

Para este caso se ha considerado la ecuación en tres dimensiones. Esta ecuación describe la distribución de calor en una región en función del tiempo.

Para resolver esta ecuación es muy importante tener en cuenta las condiciones de frontera o contorno del sistema. Estas dependen de la situación física. Como se ha planteado la expresión en tres dimensiones, supóngase que se está en un recinto de dimensiones  $(A \ B \ C)$  correspondientes a las tres dimensiones  $(x \ y \ z)$ . Entonces existirán 6 condiciones de frontera correspondientes a los extremos en cada dimensión. Es decir 0 y  $A$  para  $x$ , 0 y  $B$  para  $y$  y 0 y  $C$  para  $z$ . Las condiciones de frontera pueden ser de diferentes tipos pero generalmente son una de dos: De extremo fijo y de extremo libre. Se muestran a continuación para el extremo superior en  $x$ .

$$\begin{aligned} u(x = A) &= Cte && \leftrightarrow \text{Extremo fijo} \\ \frac{\partial u}{\partial x} \Big|_{x=A} &= Cte && \leftrightarrow \text{Extremo libre} \end{aligned} \quad (2.4.5)$$

Como puede verse, la condición de extremo fijo consiste en asignar un valor constante a  $u$  en la frontera. La condición de extremo libre es similar pero se asigna a la derivada de  $u$ . Muchas veces el valor asignado es cero. La condición de frontera de extremo fijo se conoce también como condición de Dirichlet y la condición de extremo libre como condición de Neumann.

Retomando la idea anterior, lo que se quiere es mostrar la relación entre la ecuación de calor y el filtrado gaussiano. Por simplicidad supóngase que se tiene la ecuación de calor en una dimensión:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad (2.4.6)$$

Aplicando las técnicas de los grupos de Lie (ver [51]). Se tiene que la ecuación de calor es invariante bajo las siguientes transformaciones:

- $G_1^\lambda(t, x, u) = (t, x + \lambda, u)$
- $G_2^\lambda(t, x, u) = (t + \lambda, x, u)$
- $G_3^\lambda(t, x, u) = (t, x, \lambda u)$
- $G_4^\lambda(t, x, u) = (\lambda^2 t, \lambda x, u) \quad (2.4.7)$
- $G_4^\lambda(t, x, u) = (\lambda^2 t, \lambda x, u)$
- $G_5^\lambda(t, x, u) = (t, x + 2\lambda t, u e^{-\lambda x - \lambda^2 t})$
- $G_6^\lambda(t, x, u) = \left( \frac{t}{1+4\lambda t}, \frac{x}{1+4\lambda t}, \frac{u}{\sqrt{1+4\lambda t}} e^{\frac{-\lambda x^2}{1+4\lambda t}} \right)$
- $G_7^\lambda(t, x, u) = (t, x, u + \lambda v)$

Es interesante y fácil comprobar que efectivamente cualquiera de estas transformaciones representa un grupo respecto a  $\lambda$ . A continuación se demuestra el caso de la primera transformación:

Sean  $\lambda$  y  $\gamma$  dos números reales. Se tiene que cumplir para la primera transformación que:

$$G_1^\lambda \circ G_1^\gamma = G_1^{\gamma+\lambda} \quad (2.4.8)$$

Se tiene lo siguiente:

$$G_1^\lambda \circ G_1^\gamma(t, x, u) = G_1^\lambda(t, x + \gamma, u) = (t, x + \gamma + \lambda, u) = G_1^{\gamma+\lambda}(t, x, u) \quad (2.4.9)$$

Está demostrado.

Bien. Desde el punto de vista del procesamiento digital de imágenes cada una de estas transformaciones tienen una interpretación y una utilidad que es deseable cuando se escoge el filtro que se va a aplicar. Existe una fuerte relación entre las invariantes deseadas y la solución de la ecuación de calor. Por mencionar algunas de estas características se tienen las siguientes: Las transformaciones  $G_1^\lambda$  y  $G_2^\lambda$  indican que la solución de la ecuación de calor es invariante por traslaciones, esto es deseable ya que indicará que el análisis será independiente de la ubicación de los objetos en la imagen. Las transformaciones  $G_3^\lambda$  y  $G_7^\lambda$  indican que la ecuación de calor es lineal. La transformación  $G_4^\lambda$  indica que hay una relación entre la variable espacial ( $x$ ) y la variable temporal ( $t$ ), esto equivale a que en la imagen inicial, si se hace un zoom (matemáticamente es una transformación  $x \rightarrow \lambda x$ ) se puede recuperar la solución de la ecuación mediante la transformación  $t \rightarrow \lambda^2 t$ .

Se ha mencionado la importancia del filtrado gaussiano y ciertas ventajas que tiene la ecuación de calor en el estudio de imágenes. En 1984 [37] se demostró que la convolución de una imagen dada por la función  $B_0(x, y)$  con una función gaussiana de desviación típica  $\sigma$  es equivalente a la resolución para un cierto tiempo  $t = \sigma^2/2$  de la

ecuación de calor ( $u_t = \Delta u$ ). O en palabras simples el filtrado gaussiano con todas sus ventajas es equivalente a resolver la ecuación de calor sobre la imagen.

Naturalmente esto ha generado una amplia gama de trabajos donde se utiliza la ecuación de calor como base para desarrollar determinadas técnicas o procedimientos.

## **2.5. Texturas en el procesamiento digital de imágenes.**

El propósito de esta sección es revisar algunos elementos básicos referentes a texturas y propagación de texturas haciendo énfasis en los aspectos que se tomaron para el modelo propuesto en este trabajo.

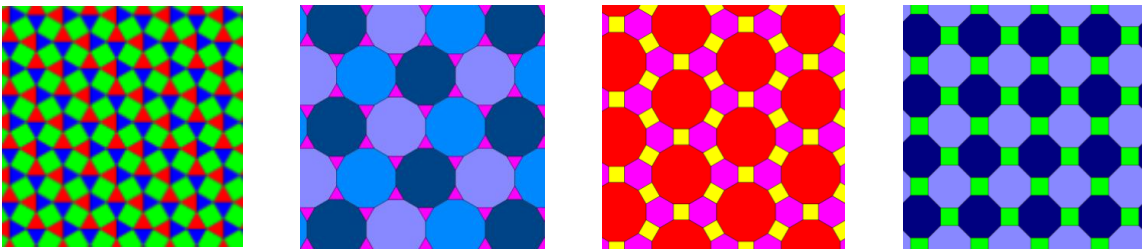
En primer lugar, es importante tener presente el concepto de texel. La palabra viene de la contracción en inglés de “texture element”. El texel es la unidad mínima de una textura aplicada a una superficie usada en gráficos por computador. [23] El texel no siempre es idéntico, en una textura determinada. Se puede decir que así como una imagen digital está constituida por una matriz de píxeles, una textura está constituida por una matriz de texeles.

El concepto de texel es muy importante cuando se hace “mapeado de texturas” que es el proceso de aplicar una textura a una superficie tridimensional. Por lo general, un texel corresponde a varios píxeles, sin embargo esto no quiere decir que dichos píxeles tengan un color uniforme. [56]

Existen métodos y algoritmos [49] para determinar el texel en una determinada imagen o textura. Este proceso se conoce como thresholding, que traduciría umbralización. Respecto de la estructura matemática de este proceso, muchos algoritmos se basan en el teselado de Voronoi. Un teselado consiste en subdividir una superficie plana de forma regular o con un patrón de figuras que cumple con las siguientes dos características:

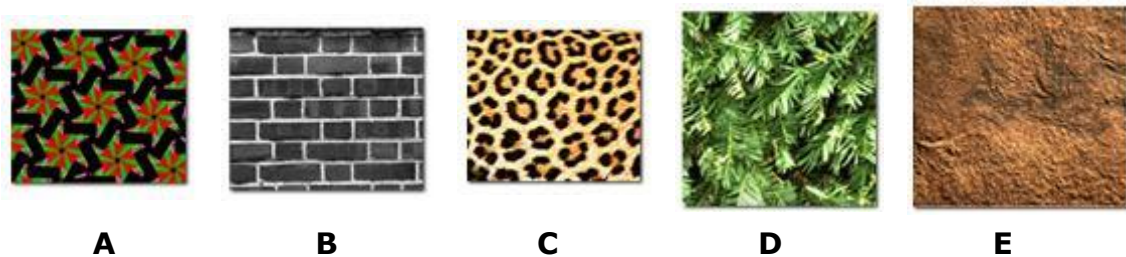
- Que no se superpongan figuras.
- Que no queden espacios vacíos.

Así el teselado puede ser muy simple, con polígonos regulares (por ejemplo cuadrados o triángulos) o puede ser un poco más complejo formándose por varias figuras y siendo así un teselado irregular. La siguiente figura muestra ejemplos de teselados semi-regulares:



**Figura 2.5-1.** Ejemplos de teselados semiregulares. Imágenes tomadas de [42]

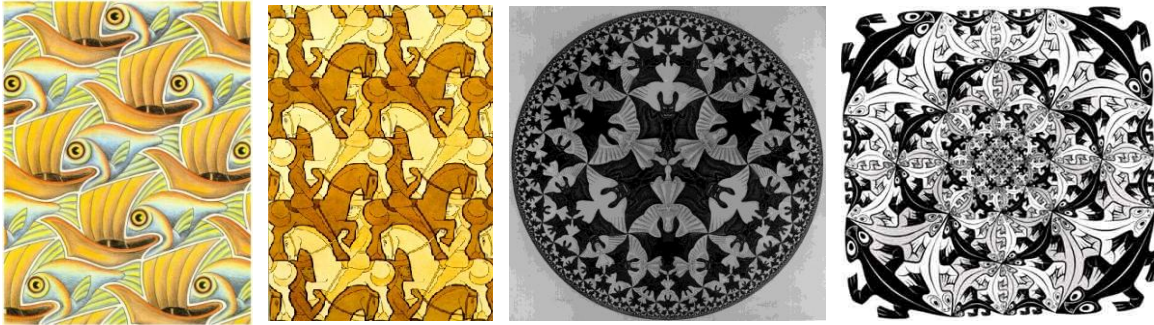
Una posible clasificación de los tipos de texturas es: Regular, semiregular, irregular, semiestocástica y estocástica. En la siguiente figura se muestra un ejemplo para cada caso:



**Figura 2.5-2.** Ejemplos de texturas: **A:** Regular, **B:** Semiregular, **C:** Irregular, **D:** Semiestocástica, **E:** Estocástica. Imágenes tomadas de [42]



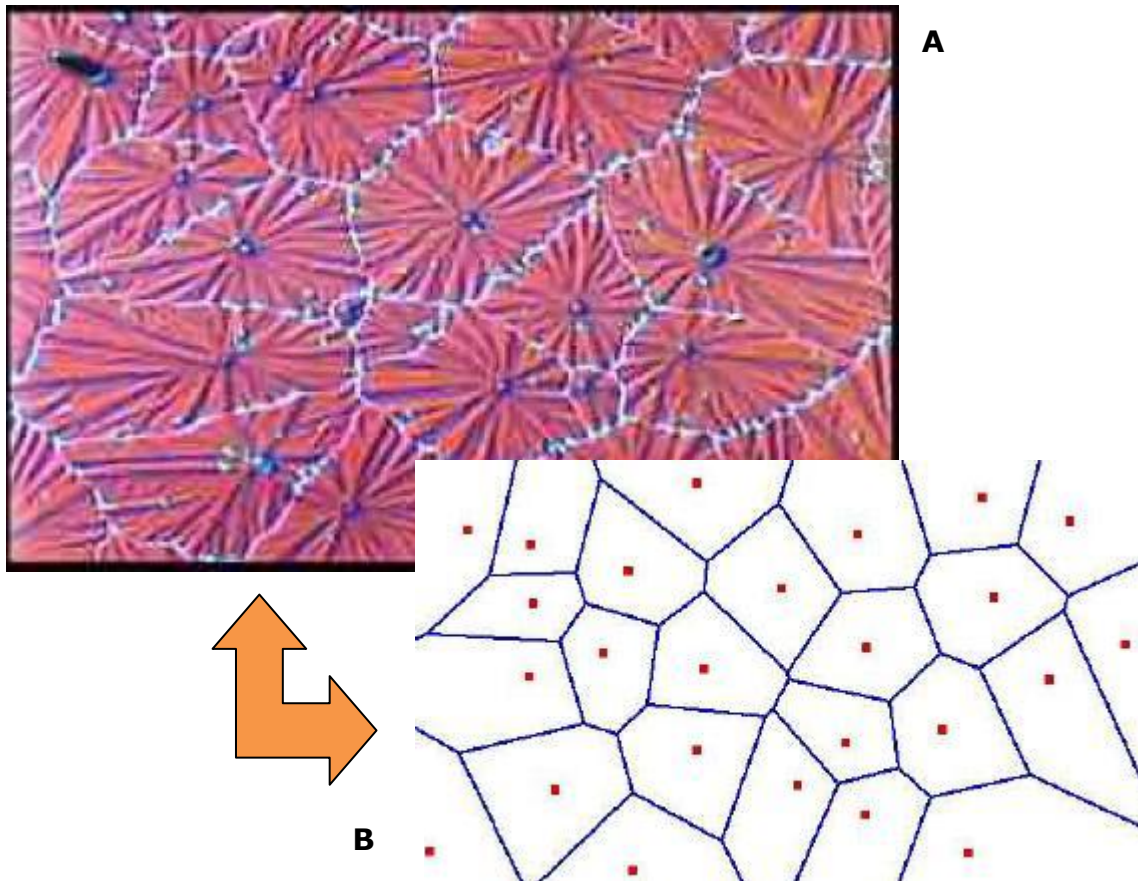
Históricamente hablando y en el campo del arte, son interesantes las obras del artista holandés Escher (1898 – 1972) [34], quien entre otras cosas, trabajó en teselados como los que se muestran a continuación:



**Figura 2.5-3.** Obras de Maurits Cornelis Escher (1898 – 1972). En las dos de la izquierda se cubre el plano cumpliendo con las características de un teselado. En las dos obras de la derecha se hace además una aproximación al concepto de infinito mediante el cambio en el tamaño de las figuras. Imágenes tomadas de [14]

Con respecto al teselado de Voronoi (también conocido como diagrama de Voronoi) basta decir que consta de los siguientes elementos: Se tiene un conjunto de puntos  $S$  en el plano que se llamarán los puntos de Voronoi. Cada punto tiene asociada una celda llamada celda de Voronoi. La celda de Voronoi se constituye por todos los puntos en el plano que están más cerca del punto de Voronoi en cuestión, que de cualquier otro punto de Voronoi. Los segmentos del diagrama de Voronoi corresponderán a los puntos del plano que se encuentran equidistantes a dos puntos de Voronoi (los dos más cercanos). Finalmente los nodos de Voronoi son puntos que se encuentran equidistantes a tres o más puntos de Voronoi. [43]

Los diagramas de Voronoi tienen diferentes aplicaciones entre las que resalta la descripción de diferentes sistemas naturales. En la figura *Figura 2.5-4* se muestra un ejemplo [55].



**Figura 2.5-4. A:** Cristal. Fotografía en escala nanométrica. **B:** Diagrama de Voronoi. El cristal se describe en su geometría, satisfactoriamente por el diagrama de Voronoi. Imágen tomada de [55]

El teselado de Voronoi tiene también aplicaciones como el estudio de propagación del cólera en epidemiología [34] y algoritmos de cuantización vectorial [71] utilizados por ejemplo en compresión de datos.

Uno de los aspectos más importantes en el trabajo con texturas es la síntesis de texturas (“texture síntesis”). Esta consiste en construir, a partir de una pequeña imagen de muestra, una gran imagen que conserve la estructura de la muestra. En otras palabras,

generar una textura a partir de una imagen. Para ello se han generado diferentes algoritmos [49][61][62]. A continuación se listan algunas familias de estos algoritmos<sup>1</sup>:

- *Embaldosamiento*: Son algoritmos simples que se basan en la repetición secuencial de la imagen generando un “embaldosado” con la misma.
- *Síntesis estocástica de textura*: Son algoritmos que mediante la generación de números aleatorios pero con algún criterio para mantener la textura, asignan colores a cada pixel.
- *Síntesis de textura con un solo propósito*: Son algoritmos muy específicos que funcionan muy bien pero solo para una textura particular. Estas texturas pueden ser por ejemplo: texturas con nubes, hojas, ladrillos, piedras, etc.
- *Mosaico caótico*: Corresponde a algoritmos similares a los del proceso *Embaldosamiento* pero la repetición no se hace con la imagen completa sino con partes aleatorias de la misma. Esto implica que en los empalmes aparezcan “costuras” las cuales son luego suavizadas. Este proceso tiene la ventaja de que la imagen producida no es siempre repetitiva. Tiene la desventaja de que el proceso de suavización puede hacer que la imagen resultante sea borrosa.
- *Síntesis de textura por pixel*: Estos algoritmos se basan en criterios de similitud. La imagen se va generando pixel por pixel. El criterio para llenar un pixel con un determinado color es examinar el resto de la imagen y escoger el pixel que tenga el entorno conocido más similar al que se quiere llenar. Una vez identificado dicho pixel, simplemente se copia su color.
- *Síntesis de textura por parches*: Estos algoritmos son similares a los anteriores pero el proceso de llenado de la imagen no se hace por pixeles sino por parches o recuadros que abarcan varios pixeles. Estos parches tienen cierto parecido con el concepto de texel aunque no es exactamente lo

---

<sup>1</sup> Se coloca el nombre traducido al español. Es más fácil encontrar información buscando el nombre en inglés.

mismo. En este caso se debe definir un criterio de similitud para comparar el parche que se va a llenar con todos los posibles parches en la imagen de muestra. La siguiente figura ilustra esta idea. En lo que resta de este documento se hará referencia a esta técnica por su nombre en inglés: *“Path based texture synthesis”*.



**Figura 2.5-5.** *Patch based texture synthesis.* A la derecha se va construyendo una imagen grande con la textura. Esta se va generando mediante parches escogidos en la imagen de muestra (izquierda). Es muy importante el hecho de que el parche no es siempre el mismo. [54]

Este tipo de algoritmo es muy importante en el presente trabajo ya que como se explicará más adelante, uno de los criterios que se tendrá en cuenta, será muy similar al implementado en esta técnica.

- *Simulación de procesos químicos:* Estos algoritmos generan texturas mediante la simulación de reacciones químicas en fluidos.

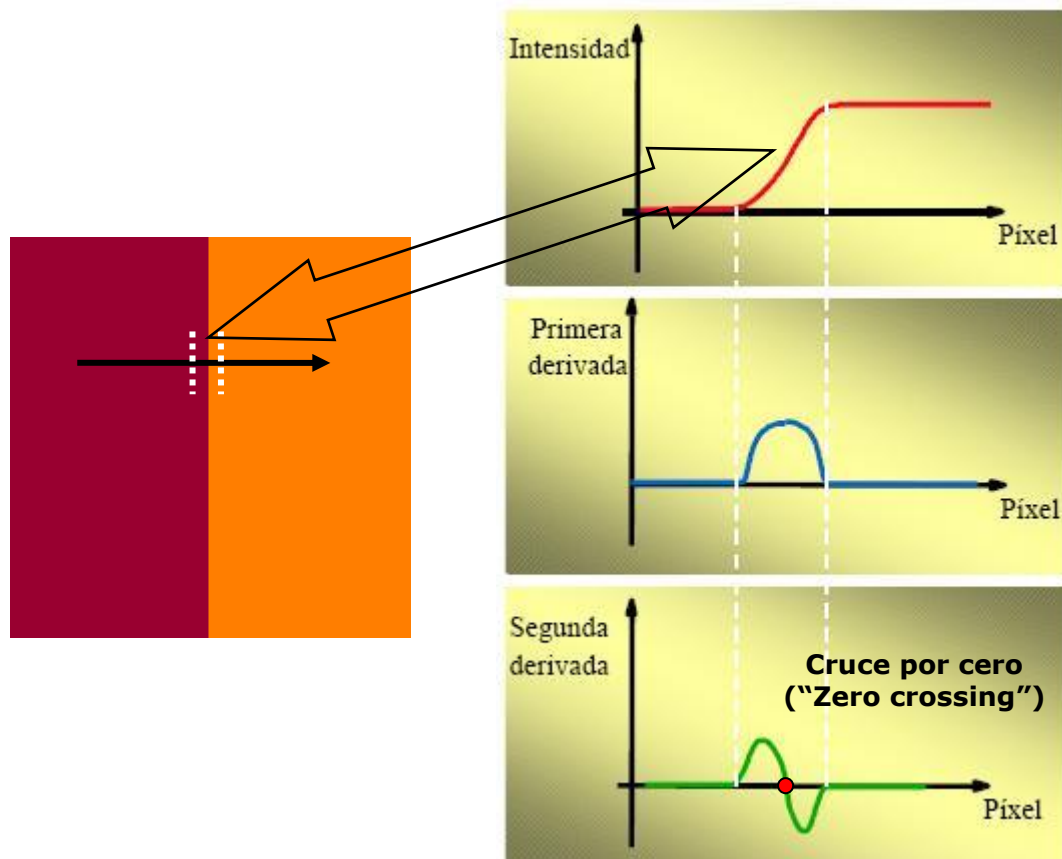
## 2.6. Detección de bordes.

En el análisis de una imagen es importante distinguir sobre la misma, los objetos que se desean estudiar, del resto de la imagen. Las técnicas que se utilizan para hacer dicha distinción se conocen como técnicas de segmentación. Una de las técnicas de segmentación típicas es la detección de bordes.

Una buena detección de bordes facilita la identificación de las fronteras de los objetos de la imagen y por lo tanto su reconocimiento.

Al referirse a bordes se puede pensar que lo que se está buscando son contornos, lo cual no es necesariamente cierto. En si el propósito de los algoritmos de detección de bordes es obtener como resultado una imagen donde se resalten los pixeles de aquellos puntos de la imagen original en donde se presentan cambios bruscos de intensidad.[10][29]

Los métodos más comunes de detección de bordes consisten en la aplicación a la imagen de filtros de suavizado seguidos de filtros de derivadas. Esto último implica simplemente que estos métodos se basan fundamentalmente en el gradiente y el laplaciano de la imagen. Desde el punto de vista del gradiente, el borde estará en donde haya grandes picos. Desde el punto de vista del laplaciano, los bordes estarán donde haya cruces por cero. Esto último ha dado lugar a la técnica de cruces por cero ("zero crossing"). La siguiente figura ilustra esta idea. [4][41]



**Figura 2.6-1.** Técnica(s) de cruces por cero ("Zero crossing"). Comportamiento de la primera y segunda derivada en un borde. Al usar filtros que implementen la primera derivada, los bordes se identificarán por los máximos y mínimos. Al tratarse de la segunda derivada los bordes se identificarán por los cruces por cero. Como se verá en el siguiente capítulo, la imagen de la izquierda fue una de las primeras que se utilizó para hacer algunas pruebas básicas con el modelo matemático propuesto en este trabajo.

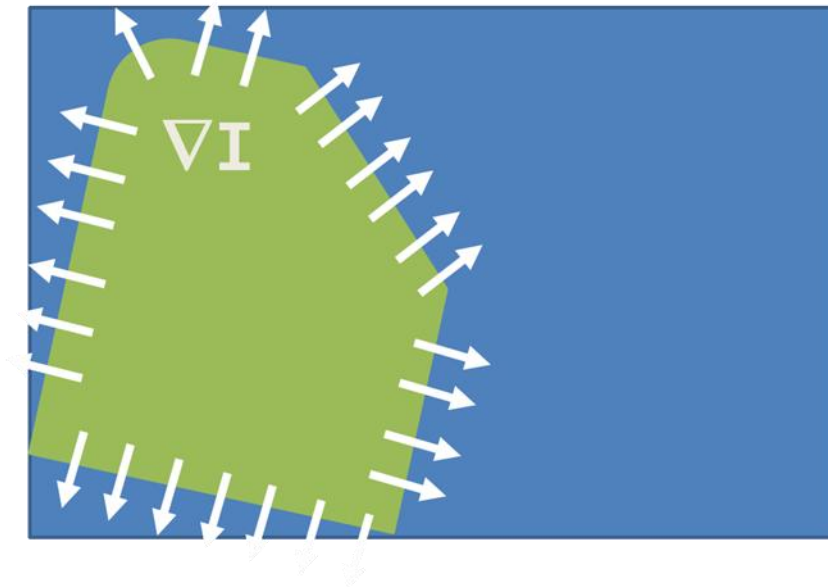
Los siguientes son algunos de los algoritmos (operadores) más conocidos para detección de bordes: [29]

- Operador de Roberts.
- Operador de Sobel.
- Operador de Prewit.
- Operador Isotrópico (o de Frei – Chen)
- Operador de Marr - Hildreth.
- Operador de Canny.

Operadores  
basados en el  
gradiente

Operadores  
basados en cruces  
por cero.

La siguiente figura ilustra el uso del gradiente para detectar bordes en una imagen.



**Figura 2.6-2.** Gradiente de una imagen simple con bordes bien definidos. Los vectores indican la dirección en la cual es mayor la razón de cambio en la intensidad de la imagen. Por lo tanto es claro que en las zonas de un solo color el gradiente es cero.

La magnitud del gradiente en cada punto (pixel) indica que tan marcado está el borde. Así, se puede decir que un pixel de borde se describe mediante dos elementos:



- Intensidad del borde. Equivale a la magnitud del gradiente.
- Dirección del borde. Es perpendicular a la dirección del gradiente.

Así los algoritmos (operadores) mencionados anteriormente utilizan mascarar determinadas para estimar el gradiente de la imagen.

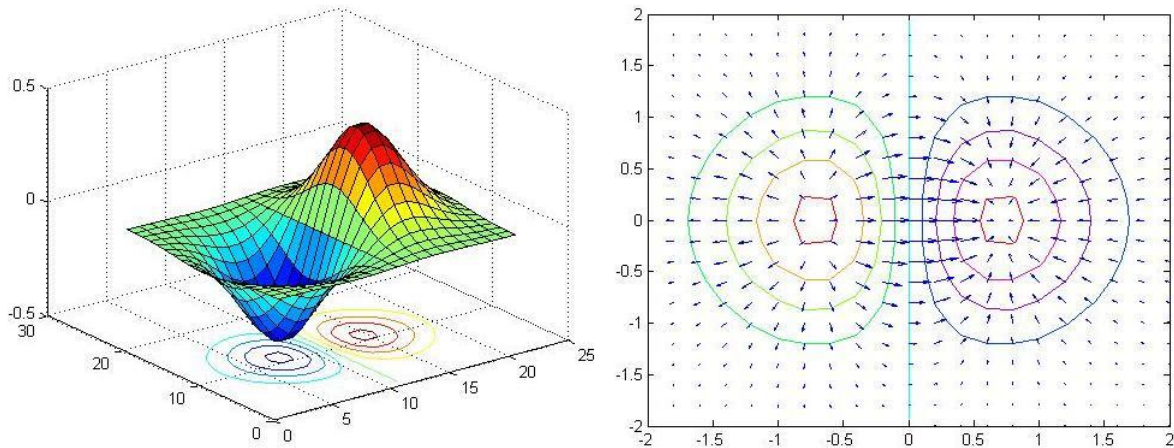
Por otro lado, en la sección 2.3 se mencionó que el filtrado de una imagen se puede hacer no solamente en el dominio del espacio sino en el de la frecuencia. Esto constituye también una técnica típica para la detección de bordes. Lo que se hace en pocas palabras es lo siguiente [41][33][24]:

- Se aplica la transformada de Fourier a la imagen obteniendo su equivalente en el espacio de frecuencias.
- A la imagen obtenida en el espacio de frecuencias se le aplica un filtro pasa alto. Este filtro lo que hace es atenuar las frecuencias bajas y mantener las altas. Puesto que las frecuencias altas corresponden a cambios bruscos de intensidad lo que se está haciendo es realzar los bordes, pues estos contienen gran cantidad de dichas frecuencias. Este filtro también realza los contrastes en la imagen.
- Finalmente se hace la transformada inversa de Fourier y se obtiene la imagen filtrada en el dominio del espacio.

Este proceso de filtrar la imagen en el espacio de las frecuencias tiene la ventaja de en general hay una buena asociación entre frecuencias bajas y cambios suaves en la tonalidad de la imagen; y frecuencias altas para cambios bruscos (bordes).

Por otro lado una técnica con diferentes aplicaciones entre ellas la de detección de bordes es la denominada "Level set" o de curvas de nivel. Para explicarla se puede tener en cuenta la siguiente figura.



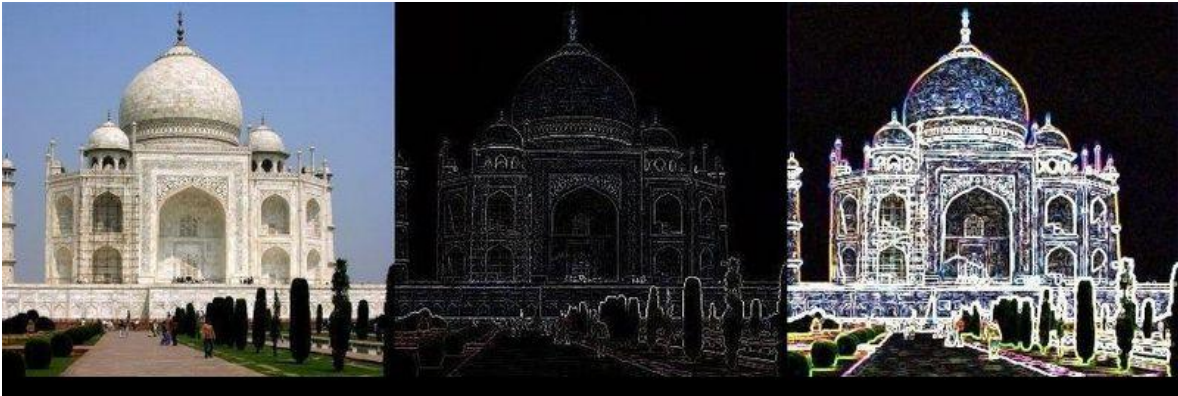


**Figura 2.6-3.** Gradiente de una función bidimensional como lo es una imagen. A la izquierda se observa la gráfica tridimensional de la función. A la derecha se observa el campo vectorial generado por el gradiente. Los vectores indican la dirección de máximo crecimiento de la función.

Lo que ilustra esta figura es el concepto de gradiente. Una imagen en sí, es una función bidimensional como la de la figura *Figura 2.6-3*. Si se toman cortes para iguales valores de intensidad se obtendrán curvas de nivel (en inglés: “Level set”) que en su conjunto serán una huella digital de la imagen, es decir son curvas únicas en cada caso. Sobre dichas curvas se pueden hacer diferentes tratamientos matemáticos que constituyen las técnicas de curvas de nivel.

Se muestran a continuación algunos resultados obtenidos mediante algoritmos relacionados con estas técnicas:

**A:**



**B:**



**C:**



**Imagen 2.6-4.** Detección de bordes. **A:** Resultado obtenido mediante el operador de Sobel (Tomado de [31]). **B y C:** Resultados obtenidos mediante el algoritmo de Canny (Tomado de [31])

### 3. Modelo matemático propuesto.

Teniendo en cuenta el marco teórico explicado, es el momento de pasar al modelo matemático. En primera instancia se explica paso a paso el modelo propuesto en este trabajo (sección 3.1). Luego se muestran los resultados y su respectivo análisis haciendo algunas observaciones particulares (sección 3.2).

#### 3.1. Estructura matemática del modelo.

Para hacer más interesante y clara la explicación del modelo propuesto en este trabajo se adelantarán algunos resultados para posteriormente mostrar con detalle como es la estructura matemática del mismo y como se obtuvieron.

Así, en la *Imagen 3.1-1* se muestra uno de los primeros resultados significativos de este trabajo. Se tiene una fotografía de la Plaza de Bolívar en Bogotá. La dimensión de la imagen es de 330 x 273 pixeles. En este caso la restauración consiste en quitar algo que no se desea que hubiera salido en la foto. Se quiere una visión de la plaza como tal, por lo cual, es necesario quitar las dos personas que aparecen. Así el usuario demarca la zona a restaurar con un color definido (*Imagen 3.1-1B*), que en este caso es un amarillo de componentes (254, 242, 0) en el espacio RGB. El programa de cómputo que implementa el modelo se ejecuta y se obtiene como resultado la *Imagen 3.1-1C* que es la imagen restaurada. Es de notar que la única dependencia con el usuario consiste en la demarcación de la zona a restaurar.

**A:**



**B:**



**C:**



**Imagen 3.1-1.** Plaza de Bolívar. Restauración mediante el modelo propuesto en este trabajo. **A:** Imagen a restaurar. **B:** Imagen demarcada. **C:** Imagen restaurada. Iteraciones: 165. Parche: 9x9 píxeles.

De esta manera lo que se hará en esta sección, es explicar paso a paso como se fue generando el modelo y cuál es su estructura matemática para llegar a este resultado, así como otros que se presentarán.

Vale la pena comentar que en el resultado de la *Imagen 3.1-1*, se ha utilizado un tamaño del parche de 9x9 píxeles y fueron necesarias 165 iteraciones. Más adelante se explicarán estos parámetros.

De acuerdo a lo desarrollado en las secciones anteriores se puede decir que hay tres elementos fundamentales a tener en cuenta para el modelo. Cada uno de estos, con un marco teórico amplio, una amplia gama de publicaciones y unas técnicas bien definidas. Estos son:

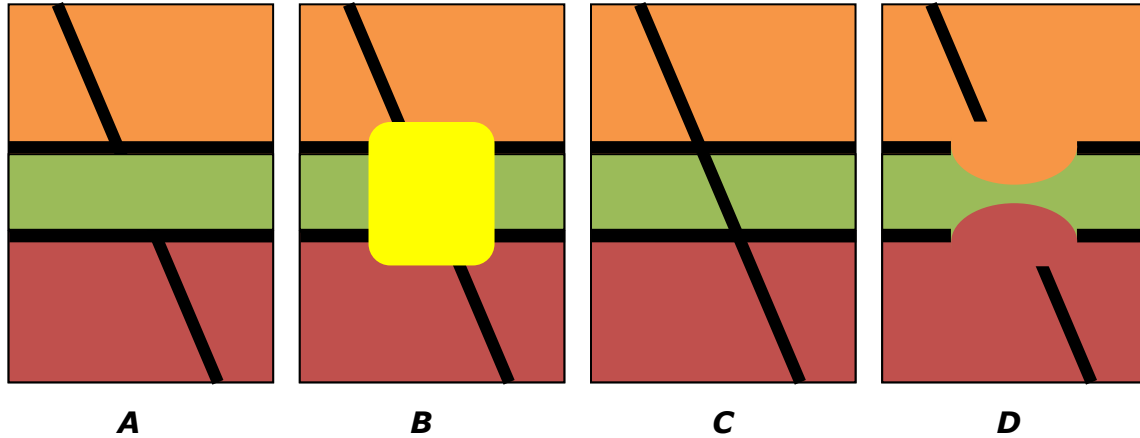
- Síntesis de texturas.
- Detección y propagación de bordes.
- Ecuación de difusión o equivalentemente filtrado Gaussiano.

El último, aunque es interesante, tiene la dificultad de que una vez aplicada la ecuación de difusión, se genera una especie de borrado o difuminado (en inglés: “blurring”) de la imagen y es necesario buscar otras técnicas (no siempre efectivas) para tratar de aclarar la imagen [32][58].

Por esta razón el modelo matemático propuesto en este trabajo se basó en los dos primeros elementos, es decir, síntesis de texturas y propagación de bordes.

En primer lugar se debe contar con que el área a restaurar está bien definida.

Algo interesante que se encontró es el hecho de que se le debe dar una “prioridad equilibrada” a la propagación de colores y de bordes respectivamente, es decir, ninguna debe primar sobre la otra. En si la propagación de colores es equivalente a lo que se está llamando propagación de textura, mientras que la propagación de bordes es lo que se denominará, propagación de estructura, es decir lo que define los objetos en la imagen. Para aclarar lo que se quiere decir con una “prioridad equilibrada” es útil la siguiente figura:



**Figura 3.1-1.** Ejemplo intuitivo de una posible restauración. **A:**Imagen original. **B:** Imagen deteriorada. **C:** Imagen restaurada dando alta prioridad a la propagación de bordes (estructura). **D:** Imagen restaurada dando alta prioridad a la propagación de colores (textura).

Esta figura pretende dar una primera explicación intuitiva de la forma en que compiten la propagación de textura y de estructura en la zona a restaurar de la imagen. En la figura *Figura 3.1-1A*, se tiene la imagen original o correcta, la *Figura 3.1-1B* es la imagen a restaurar, es decir, se supone que el recuadro interno (en color amarillo) es una mancha o algo que afectó la imagen e impide saber cómo era. La *Figura 3.1-1C* es una versión intuitiva de lo que se obtendría posiblemente, si se restaura la imagen dando total prioridad a la propagación de bordes. En este caso se propagan primero los bordes en el área de restauración y una vez que se ha hecho, se rellenan los espacios con los colores respectivos de la frontera. La *Figura 3.1-1D* es una versión intuitiva de lo que se obtendría posiblemente, si se restaura la imagen dando total prioridad a la propagación de colores. En este caso simplemente se propagan los colores dejando los bordes intactos. Es claro que ni la *Figura 3.1-1C* ni la *Figura 3.1-1D* corresponden a la “solución correcta”, o sea la *Figura 3.1-1A*. Es necesario entonces proponer un modelo matemático que tenga en cuenta los dos elementos, a saber, la propagación de texturas y de estructura de una manera unificada y equilibrada.

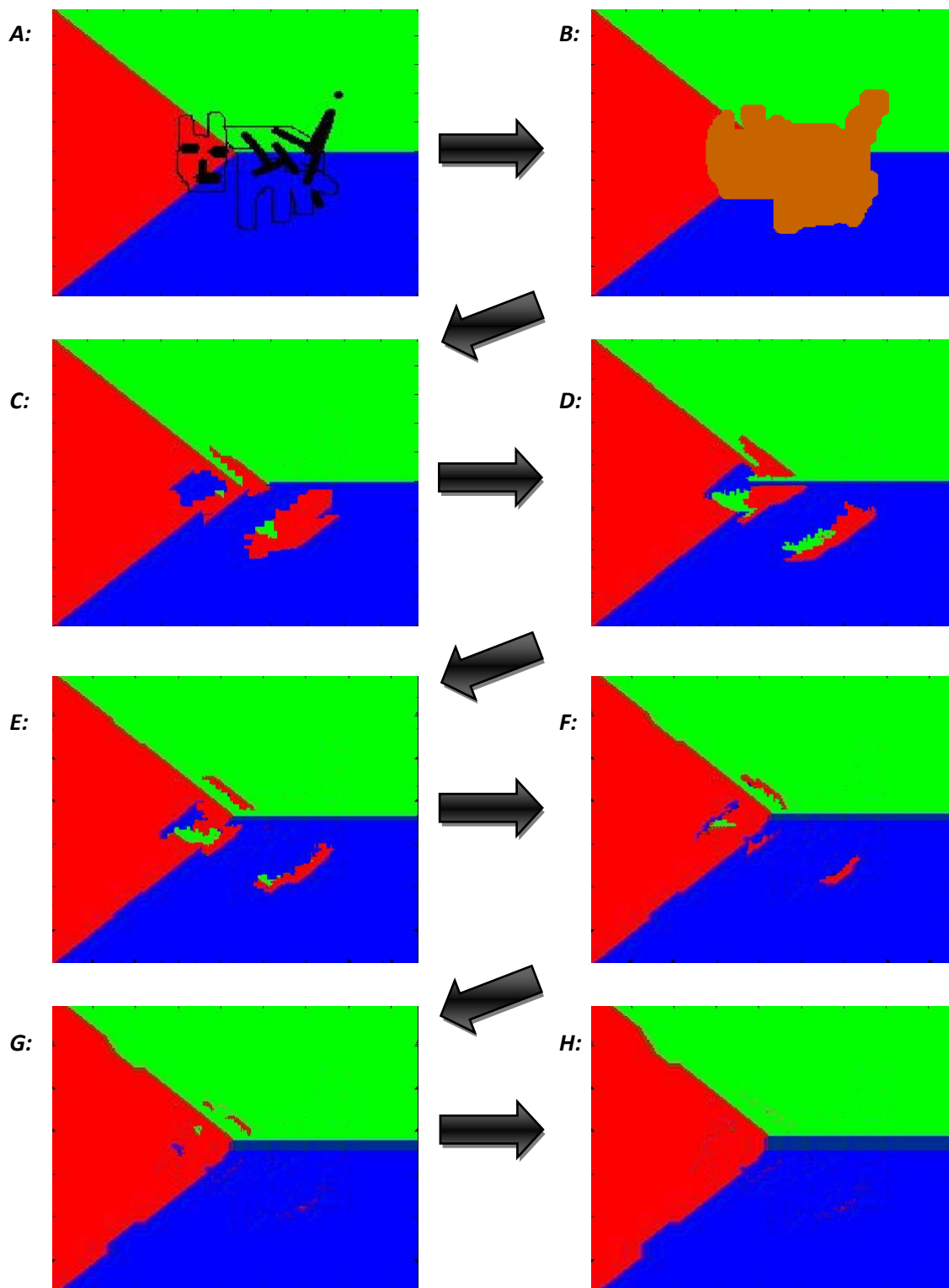
Antes de continuar es bueno tener en cuenta un aspecto del razonamiento anterior. La *Figura 3.1-1* tiene como objetivo explicar el presente modelo, sin embargo es claro que



al momento de restaurar una imagen nunca se sabrá cual es la solución correcta, es decir “nunca se tendrá la *Figura 3.1-1A*”. Al momento de restaurar una imagen, se hace porque está deteriorada y no se sabe exactamente como era originalmente. En este sentido el problema que aborda este trabajo es abierto y no se puede hablar de una solución exacta. Lo que se espera es que los resultados que arroje el modelo, es decir las imágenes restauradas sean visualmente satisfactorias. Así, se busca que la restauración sea indetectable en un visor común, por el usuario que desconoce que la imagen se ha procesado.

La *Figura 3.1-1* es simplemente un dibujo para facilitar la explicación. Vale la pena mostrar algunos resultados intermedios o experimentales que se obtuvieron al tratar de desarrollar el modelo. En sí, para lograr consolidar una propuesta matemática lo que se hizo fue hacer pequeños “experimentos” con imágenes sencillas y estudiar los resultados para posteriormente aplicar el modelo con imágenes más complejas (ver objetivos, sección 1).

Así, uno de estos experimentos es el que se muestra en la *Imagen 3.1-2*. El tamaño de la imagen es de 433 x 340 píxeles. En este caso se utilizó la técnica “*Patch based texture synthesis*” la cual fue explicada muy puntualmente en la sección 2.5. El tamaño del parche (se aclarará más adelante) en este caso fue de 9x9 píxeles. La *Imagen 3.1-2A* es la imagen a restaurar, la *Imagen 3.1-2B* es la imagen con la zona demarcada (en este caso con un color de componentes 200,100 y 0 en el espacio de colores RGB). La *Imagen 3.1-2C* es la imagen restaurada. Como se ve, el resultado obtenido no es del todo satisfactorio por lo menos respecto de lo que se esperaría intuitivamente. Es claro que un modelo centrado solo en propagación de texturas, puede tener inconvenientes al momento de tener en cuenta los bordes. Sin embargo puede surgir la pregunta: ¿Qué pasa si se hace una restauración de la imagen restaurada? Se podría volver a demarcar la imagen restaurada y repetir el proceso. Se observó que la imagen mejora gradualmente si se restaura iterativamente. En la *Imagen 3.1-2H* se puede ver un resultado satisfactorio después de ejecutar el programa de cómputo 16 veces. El número total de iteraciones para obtener dicho resultado fue de 1638, prácticamente 10 veces más que las iteraciones necesarias para obtener la *Imagen 3.1-1*.

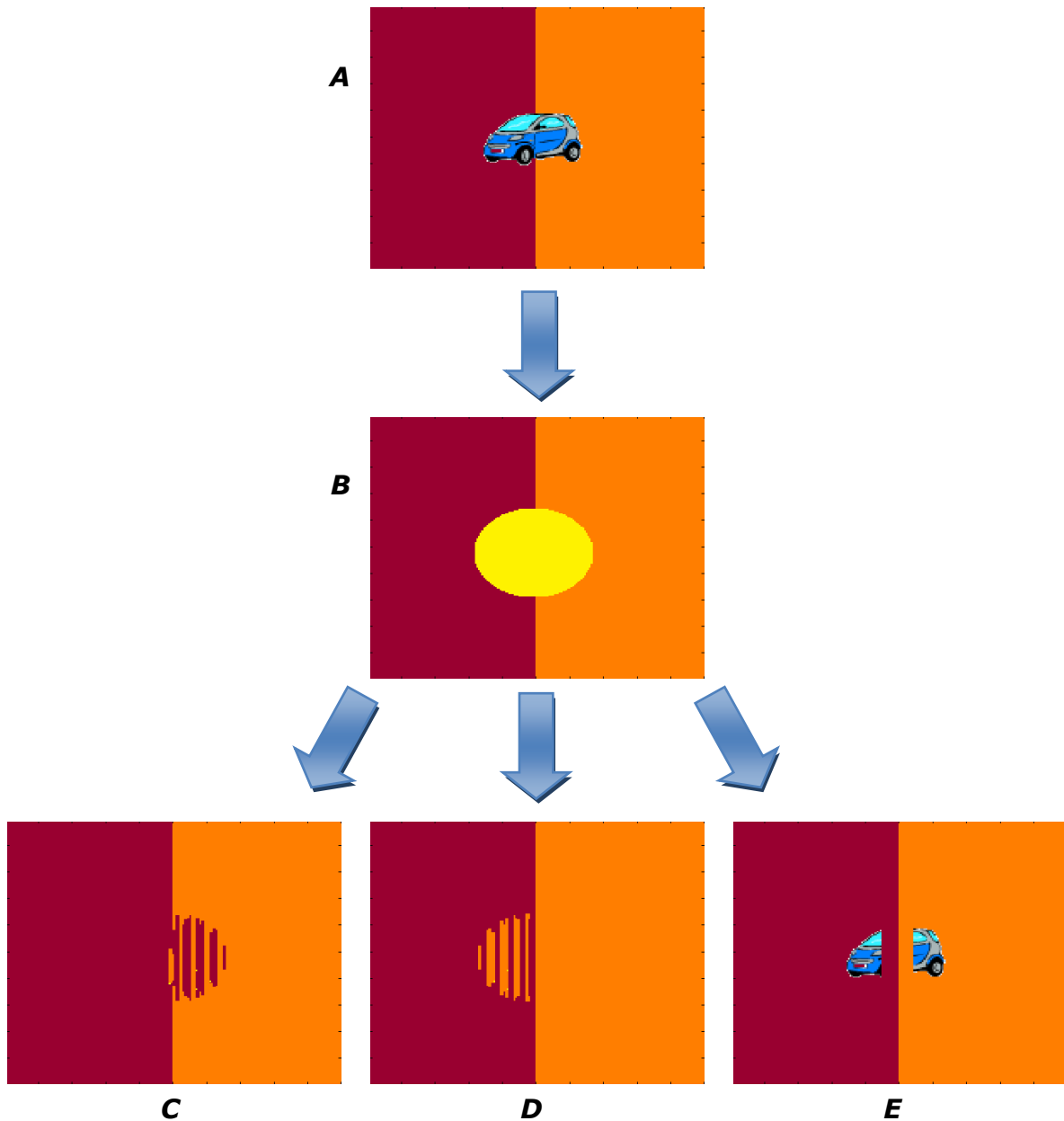


**Imagen 3.1-2.** Restauración de una imagen (primer experimento). **A:** Imagen a restaurar. **B:** Imagen demarcada. **C:** Restauración 1. **D:** Restauración 2. **E:** Restauración 4. **F:** Restauración 7. **G:** Restauración 11. **I:** Restauración 16. Iteraciones: 1638



Esta restauración se ha hecho un poco “a la fuerza”, al tener que realizar sistemáticamente una restauración tras otra. Es claro que el número de iteraciones es relativamente alto. Se necesita tener en cuenta en el modelo, la propagación de bordes, de una manera más significativa.

En este orden de ideas, conviene mostrar un segundo experimento que se hizo. Es el que se muestra en la siguiente figura:



**Imagen 3.1-3.** Restauración de una imagen (segundo experimento). **A:** Imagen a restaurar. **B:** Imagen demarcada. **C:** Restauración barriendo la imagen de izquierda a derecha. **D:** Restauración barriendo la imagen de derecha a izquierda. **E:** Restauración propagando solo el borde.

En este caso se ha tomado una imagen aún más simple, que consta de un solo borde y dos colores. Se obtuvieron tres resultados aplicando criterios diferentes. La *Imagen 3.1-3C* es el resultado de aplicar el mismo modelo ("*Patch based texture synthesis*") de la *Imagen 3.1-2*. Sin embargo, al hacer algunas pruebas se obtuvo la *Imagen 3.1-3D*. Este resultado se obtuvo aplicando exactamente el mismo modelo. Lo único que cambia es el orden en que se llena el área a restaurar. En la *Imagen 3.1-3C* el barrido por los píxeles se hizo de izquierda a derecha, mientras que en la *Imagen 3.1-3D* se hizo de derecha a izquierda. La diferencia en los resultados se puede explicar de la siguiente manera para la *Imagen 3.1-3C*: Al hacer el barrido de izquierda a derecha, se va llenando la zona a restaurar con el mismo color del entorno (vino tinto). Esto porque el criterio de comparación para reconocer la textura identifica simplemente a la misma como el color en mención. Sin embargo, al llegar al borde la textura (ver el concepto de texel en el sección 2.5) se identifica como la mezcla de los dos colores (vino tinto y naranja) por lo tanto se sigue propagando una mezcla de los dos. Para la *Imagen 3.1-3D* se tendrá el mismo razonamiento solo que haciendo el barrido en la dirección contraria. Es claro que los dos resultados son muy diferentes. Esto plantea otro elemento a tener en cuenta en el modelo: El orden en que se llene el área de restauración es muy importante.

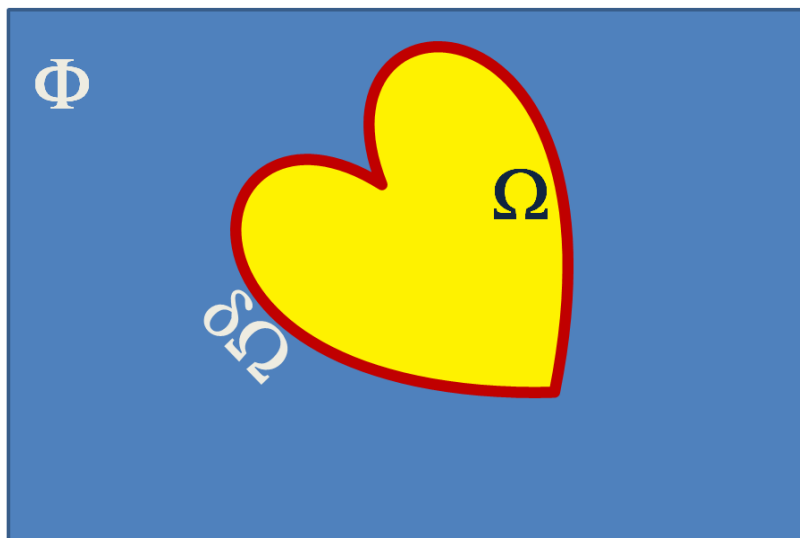
Por otro lado se obtuvo un tercer resultado correspondiente a la *Imagen 3.1-3E*. En este caso lo que se hizo fue propagar solamente el borde. En sí, se incluyó un cambio simple sobre el modelo anterior. Primero se obtuvo el gradiente de la imagen. Para un borde bien definido dicho gradiente es perpendicular al mismo (ver por ejemplo la *Figura 2.6-3*), por lo tanto es suficiente rotar el gradiente 90 grados para definir la dirección en que se quiere ir llenando (restaurando) la imagen. Así, lo que se hizo fue aplicar el mismo modelo ("*Patch based texture synthesis*"), solo que obligando a que la imagen se llene en un determinado orden (por donde "debe ir" el borde). De aquí se concluye otro aspecto importante: Si se introducen las modificaciones necesarias se pueden usar modelos de propagación de texturas para propagar bordes.

Es claro que los tres resultados son diferentes entre sí y que ninguno corresponde a lo que la intuición esperaría. Sin embargo hay que insistir en algo que ya se mencionó: El problema es abierto. Ninguno de los tres resultados es correcto pero tal vez ninguno es falso. No se tiene porque saber de antemano el resultado correcto, lo importante es que sea visualmente convincente. De hecho, para la imagen en cuestión, el resultado que

predice la intuición (dos rectángulos contiguos con cada uno de los colores) no es necesariamente el correcto.

Bien, hasta el momento se han adelantado algunos resultados (*Imagen 3.1-1*, *Imagen 3.1-2* e *Imagen 3.1-3*) sin explicar en detalle cómo se obtuvieron, pero con el ánimo de mostrar de antemano algunos elementos importantes en el modelo propuesto en este trabajo. A continuación se hace la explicación formal:

En primera instancia se utilizará una notación que ya se ha vuelto estándar en las publicaciones sobre restauración ("inpainting", ver por ejemplo [6]). Esta se ilustra en la siguiente figura:



**Figura 3.1-2.** Nomenclatura usual en restauración de imágenes (inpainting).

Se tiene entonces:

$I$  = Imagen.

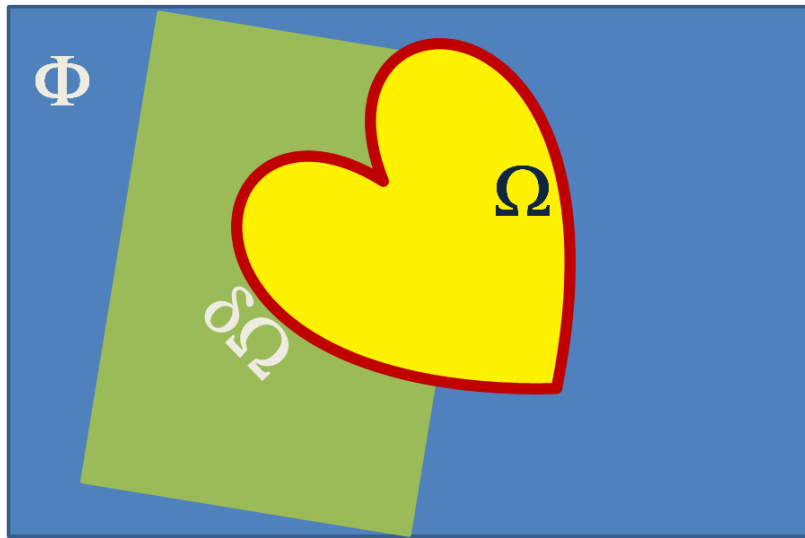
$\Phi$  = Región fuente. Es la región conocida de la imagen.

$\Omega$  = Región objetivo. Es la región a restaurar.

$\delta\Omega$  = Frontera

La forma de corazón de  $\Omega$  se ha escogido a propósito ya que facilita algunos puntos de la explicación.

En primer lugar conviene recordar el método de síntesis de texturas explicado en el sección 2.5 “*Patch based texture synthesis*” y aprovechar para mostrar al mismo tiempo cómo se puede ajustar esta técnica para propagar no solo textura sino estructura (bordes). Considérese que la imagen a restaurar es la siguiente:

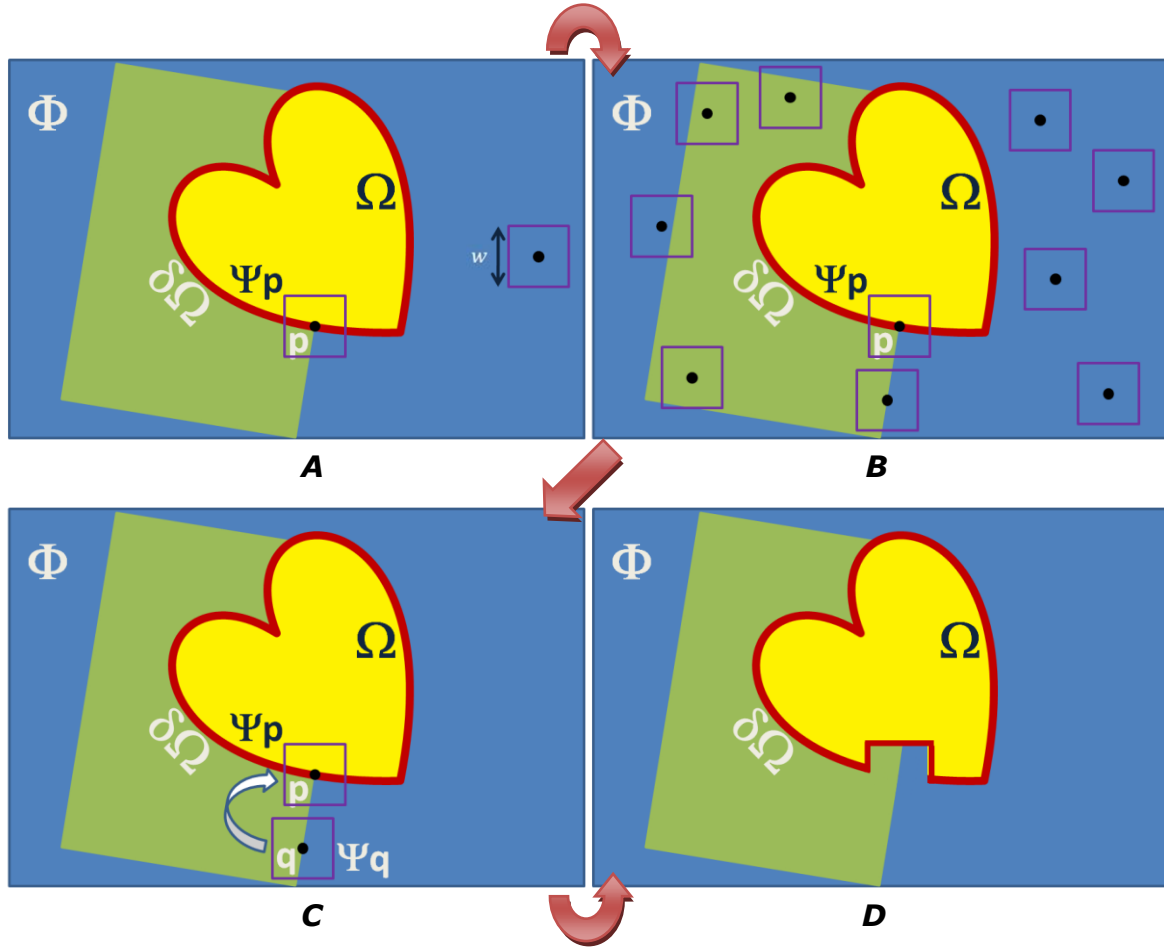


**Figura 3.1-3.** Imagen a restaurar mediante síntesis de texturas.

Como se observa, es la misma imagen de la *Figura 3.1-2* pero se ha aumentado un poco su complejidad colocando otro color en la imagen conocida  $\Phi$ .

El proceso de restauración se hace de la siguiente manera: En primer lugar se escoge un punto sobre la frontera, es decir  $p \in \delta\Omega$ . Desde dicho punto se comienza a restaurar la imagen, o en otras palabras en dicho punto se hará la primera iteración en el proceso de restauración. Supóngase que se ha escogido el punto que se muestra en la *Figura 3.1-4.A*. Nótese que para fines de la explicación se ha escogido un punto donde se presente un borde, es decir un límite entre dos colores en  $\Phi$ . El siguiente paso es definir un parche o recuadro centrado en  $p$ . Este parche abarca un número determinado de píxeles según su tamaño. Supóngase por ejemplo que el parche de 9x9 píxeles, esto quiere decir, que desde el punto  $p$ , se genera el parche dando un margen de 4 píxeles en

las cuatro direcciones (derecha, izquierda, arriba y abajo). Se utilizará la letra  $w$  para referirse al lado del parche, es decir, que el tamaño del parche será  $w \times w$ . Es importante insistir, en que siempre que se utilice la letra  $p$  para un punto o pixel de la imagen en este documento, se entenderá que dicho punto se encuentra sobre la frontera ( $p \in \delta\Omega$ ). En este orden de ideas el respectivo parche se notará como  $\Psi_p$ .



**Figura 3.1-4.** Restauración mediante síntesis de texturas (primera iteración).

El siguiente paso (ver sección 2.5) es hacer un barrido por todos los puntos de la imagen conocida  $\Phi$  y estudiar el parche asociado a cada punto. Para hacer referencia a estos puntos se utilizará la letra  $q$ . De esta manera el respectivo parche se notará  $\Psi_q$ . La idea es encontrar el parche  $\Psi_q$  que sea más parecido a  $\Psi_p$ . Esto teniendo en cuenta que la comparación se hace para los puntos  $q$  de  $\Psi_p$  que pertenecen  $\Phi$  es decir para:

$$\forall q \in (\Psi_p \cap \Phi) \quad (3.1)$$

El criterio de similitud para hacer la comparación tiene que ser de tipo matemático. Se utilizará el error cuadrático medio (MSE). Este criterio se utiliza en contextos de tipo experimental para expresar la incertidumbre de una medida de tipo aleatorio, pero también se utiliza como un criterio de proximidad entre dos juegos de datos. El error cuadrático medio está dado por la siguiente expresión:

$$\Delta xy = \sqrt{\frac{\sum_{i=1}^N (x_i - y_i)^2}{N(N-1)}} \quad (3.2)$$

Donde  $N$  es el número de datos y  $x_i$  e  $y_i$  son los dos juegos de datos. Como se observa por la forma de la expresión, esta definición está basada en el concepto de desviación estándar. Entre más pequeño sea el valor  $\Delta xy$ , mas cercanos están los datos.

Más formalmente, lo anterior se escribiría de la siguiente manera en este contexto:

$$\Delta q\tilde{q} = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (Iq_{ij} - I\tilde{q}_{ij})^2}{w^2(w^2 - 1)}} \quad \forall q \in (\Psi_p \cap \Phi) \wedge \tilde{q} \in \Psi_q \quad (3.3)$$

Donde  $Iq_{ij}$  es la intensidad o brillo en la escala RGB del punto  $q$  que pertenece a  $\Psi_p$ , de coordenadas  $ij$ .  $I\tilde{q}_{ij}$ , es la respectiva intensidad o brillo en la escala RGB del punto  $q$  que pertenece a  $\Psi_q$ , de coordenadas  $ij$ . Es decir que se está usando el símbolo  $\sim$  para diferenciar si el punto pertenece a  $\Psi_p$  o a  $\Psi_q$ . Téngase en cuenta que la letra  $p$  se utiliza únicamente para puntos que pertenecen a  $\Phi$  (y están centrados en  $\Psi_p$ ).

Una vez que se ha hecho todo el barrido sobre  $\Phi$  para estudiar todos los posibles parches  $\Psi_q$  (Figura 3.1-4.B), se escoge el que tenga mayor similitud con  $\Psi_p$  (Figura 3.1-4.C). Finalmente se restaura la zona desconocida de  $\Psi_p$ , es decir  $\Psi_p \cap \Omega$ . Para ello simplemente se reemplazan los puntos equivalentes de  $\Psi_q$  en  $\Psi_p$  (Figura 3.1-4.D). Así, se ha hecho la primera iteración del algoritmo. Es necesario escoger un nuevo

punto  $p$  y volver a empezar. Nótese el hecho de que se ha iniciado la propagación de un borde.

Se ha explicado el proceso mediante la escogencia conveniente de un punto  $p$  que coincida con un borde. Sin embargo ¿va a ser este siempre el procedimiento en todas las iteraciones? Claro que no. La forma como se escoge el punto  $p$  y en consecuencia el respectivo parche  $\Psi_p$  debe ser apropiada. Como se explicó anteriormente (ver en especial la *Imagen 3.1-3*) el orden en que se llene la zona a restaurar  $\Phi$ , incide significativamente en el resultado. También se ha hablado de la importancia de lograr un equilibrio adecuado entre la propagación de textura (colores) y de estructura (bordes) (ver la *Figura 3.1-1*).

En este orden de ideas se propuso el siguiente modelo:

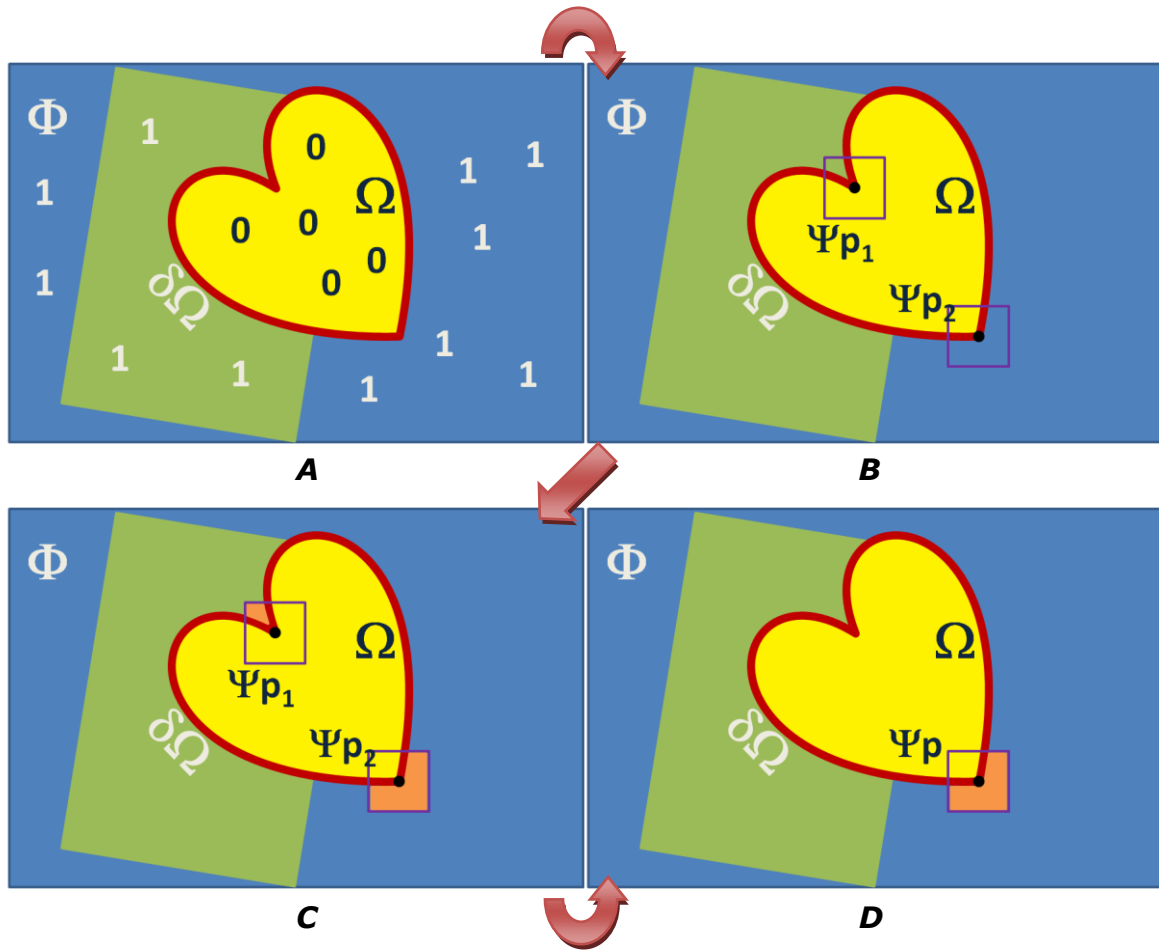
Teniendo en cuenta el conjunto de puntos  $p$  que pertenecen a  $\delta\Omega$  ( $p \in \delta\Omega$ ) se asocia una función o vector de probabilidad de la siguiente manera:

$$P(p) = C(p) \cdot B(p) \quad \forall p \in \delta\Omega \quad (3.4)$$

Donde  $C(p)$  es el vector de probabilidad de cada punto  $p$  de ser escogido según su grado de confianza y  $B(p)$  es el vector de probabilidad de cada punto  $p$  de ser escogido según su importancia para propagar un borde. En otras palabras  $C(p)$  es la probabilidad de cada punto  $p$  de ser escogido por su importancia para propagar textura y  $B(p)$  es la probabilidad de cada punto  $p$  de ser escogido por su importancia para propagar estructura. Por simplicidad se les llamará vector (o matriz) de confianza ( $C(p)$ ) y vector de bordes ( $B(p)$ ).

Se explica a continuación la forma de definir  $C(p)$ . Téngase en cuenta la *Figura 3.1-5*.





**Figura 3.1-5.** Probabilidad de un punto  $p$  de ser escogido para iniciar la restauración según su importancia para propagar textura (colores).

Para la primera iteración se define una matriz  $C$ , del mismo tamaño que la imagen. Se inicia asignando a los píxeles de la imagen conocida ( $\Phi$ ) el valor de 1 en la matriz. Esto quiere decir que dichos píxeles o puntos tienen un nivel de confianza del 100% ya que forman parte de la imagen conocida. A los píxeles de la zona a restaurar ( $\Omega$ ) se les asigna el valor 0. Es decir que su nivel de confianza es nulo, justamente porque son los píxeles a restaurar (ver Figura 3.1-5A). Se tiene:

$$C(q) = \begin{cases} 0 & \forall q \in \Omega \\ 1 & \forall q \in \Phi \end{cases} \quad (3.5)$$

La anterior definición se refiere a puntos que pertenecen a  $\Phi$  o a  $\Omega$ . Para los puntos de la frontera ( $\delta\Omega$ ) se tiene:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \Phi} C(q)}{|\Psi_p|} \quad \forall p \in \delta\Omega \quad (3.6)$$

Es decir, para los puntos  $\square$ , su probabilidad de ser escogidos por su nivel de confianza (posibilidad de propagar textura) es igual a la sumatoria de los valores  $\square(\square)$  (ver ecuación 3.5) de los puntos  $\square$  que pertenecen al respectivo parche ( $\Psi_p$ ) y se encuentran en la zona conocida ( $\Phi$ ) de la imagen. Este valor se normaliza dividiendo por un factor de normalización  $|\Psi_p|$  el cual va a ser igual al tamaño del parche. Es decir si el parche es un cuadrado de 9x9 pixeles, entonces  $|\Psi_p| = 81$ . En la notación explicada anteriormente (ver *Figura 3.1-4.A*) es claro que:

$$|\Psi_p| = w \times w = w^2 \quad (3.7)$$

Es de notar que en sí,  $C$  es una matriz del mismo tamaño que la imagen. Sin embargo cuando se evalúa solo sobre los puntos  $p$  (ecuaciones 3.4 y 3.6) se puede considerar simplemente como un vector. Por esto en los anteriores párrafos se ha referenciado de las dos maneras. Como se mostrará más adelante no sucede lo mismo con  $D$  que se debe considerar solo como un vector, pues se define solamente para los puntos  $p$ .

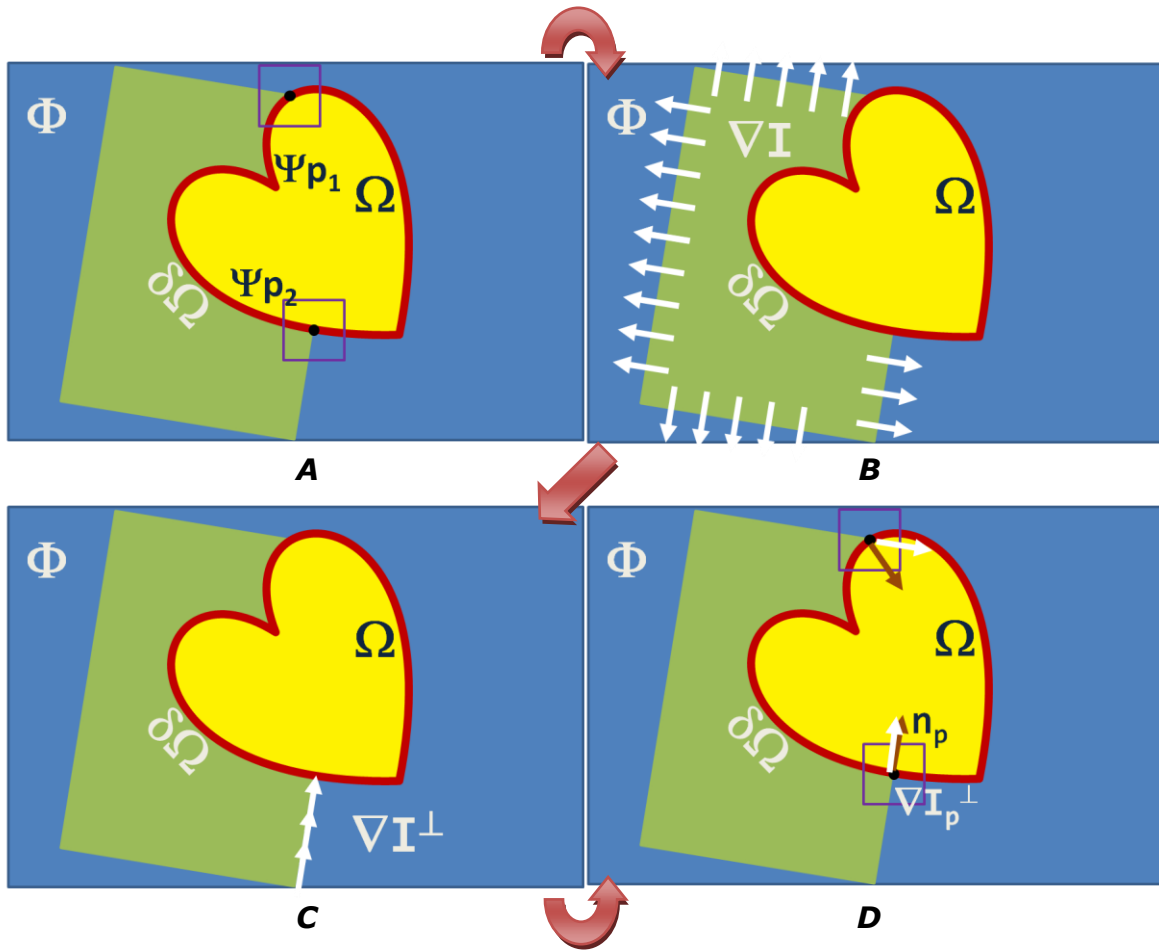
La definición dada por la ecuación 3.6 puede ser comprendida más claramente con ayuda de la *Figura 3.1-5*. Supóngase que se van a estudiar los dos parches que se muestran (*Figura 3.1-5B*). Es claro que el parche que se encuentra en la parte inferior derecha ( $\Psi_{p_2}$ ) tiene una porción mayor de “imagen conocida” o en otras palabras una mayor intersección con  $\Phi$ , como se muestra en la región sombreada (*Figura 3.1-5C*). Por lo tanto de entre los dos, es este el que se debe escoger, pues posee una mayor información de “zona conocida” en la imagen y en consecuencia de la posible textura a propagar. Claramente, la ecuación 3.6 arroja un valor mayor de  $C$  para este parche, por lo tanto es el que se escoge (*Figura 3.1-5D*). Se han mostrado estos dos parches para

facilitar la explicación pero en sí, la ecuación 3.6 se evalúa para todos los puntos  $p$  sobre la frontera ( $\delta\Omega$ ).

Bien. Se explica a continuación la segunda parte en la definición de la probabilidad  $P$  (ver ecuación 3.4), es decir  $D(p)$ , que es el vector de probabilidad para cada punto  $\square$ , de ser escogido por su importancia para propagar estructura (bordes). La definición está dada según la siguiente ecuación:

$$B(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha} \quad (3.8)$$

Para comprender esta expresión, téngase en cuenta la *Figura 3.1-6*. De nuevo se van a analizar dos parches que permitan ilustrar los conceptos que se van a tener en cuenta (*Figura 3.1-6A*). Como se observa, se han escogido dos puntos  $p$ , justamente donde llega un borde a la frontera de la región ( $\Omega$ ) a restaurar.



**Figura 3.1-6.** Probabilidad de un punto  $p$  de ser escogido para iniciar la restauración según su importancia para propagar estructura (bordes).

El primer paso es obtener el gradiente ( $\nabla I$ ) de la imagen conocida, es decir, de  $\Phi$ . Se obtendría lo que se muestra en la *Figura 3.1-6B*. Es decir, en las zonas de un solo color el gradiente es cero porque no hay variación en la intensidad de cada pixel. El gradiente se hace diferente de cero en los bordes (ver también *Figura 2.6-2.* y *Figura 2.6-3*). Considérese el punto  $p_2$ . Si en dicho punto se rota el gradiente 90 grados, se obtendrá la dirección en que se debe propagar el borde en la zona a restaurar ( $\Omega$ ) (*Figura 3.1-6C*). Esto se puede hacer también para  $p_1$  y para todos los puntos  $p$  sobre  $\delta\Omega$ . De esta manera, se tendrá en cada punto un vector dado por  $\nabla I^\perp$ . Por otro lado se puede definir el conjunto de vectores  $n$  para cada punto  $p$ , que son perpendiculares a la frontera  $\delta\Omega$  y están dirigidos hacia adentro. Es de notar que dicho conjunto constituye también un

campo vectorial pero es diferente al gradiente de la imagen. De esta manera se tienen, para cada punto  $p$ , dos vectores  $\nabla I^\perp$  y  $n_p$ .

Bien. ¿Cuál de los dos puntos  $p_1$  o  $p_2$  es más importante para propagar su respectivo borde? Intuitivamente se puede decir que  $p_2$  ya que dicho borde atravesaría la región  $\Omega$  generando una estructura importante mientras que para  $p_1$  su respectivo borde escasamente bordea la región. El criterio matemático a seguir es dar una mayor probabilidad de propagación a los bordes que sean perpendiculares a la frontera ( $\delta\Omega$ ). Para ello simplemente se hace el producto punto entre los dos vectores mencionados (ecuación 3.8) y se divide por un factor de normalización que para efectos prácticos puede ser el mismo que se utilizó en la ecuación 3.6. Es decir, se tiene:

$$\alpha = |\Psi_p| = w \times w = w^2 \quad (3.9)$$

Es claro que este criterio no siempre será correcto, es decir, no siempre un borde que sea perpendicular a la frontera será más importante que otro que no lo sea, sin embargo, es plausible considerar que casi siempre va a ser así.

Bien, finalmente se hace la restauración considerando el punto  $p$  que haya obtenido una mayor probabilidad  $P$ .

Nótese que una vez generado el programa de cómputo, solo quedan dos parámetros a disposición del usuario: En primer lugar la demarcación de la zona  $\Omega$  a restaurar ya que se pueden obtener resultados diferentes según la zona que se seleccione. En segundo lugar el tamaño del parche  $w^2$ , pues este se puede variar. Nótese de antemano que un valor “grande” de  $w$  permite obtener una información más global y macroscópica, es decir, permite obtener más datos acerca de la posible textura y estructura a propagar. Sin embargo, implica también la mayor notoriedad de posibles errores o resultados no deseados. Un valor “pequeño” de  $w$  permite hacer el proceso de una manera más fina y detallada pero disminuye la calidad de la información que se tendrá al momento de hacer la restauración de cada parche  $\Psi_p$ .

El siguiente esquema muestra brevemente como es el programa de cómputo que implementa el modelo explicado.

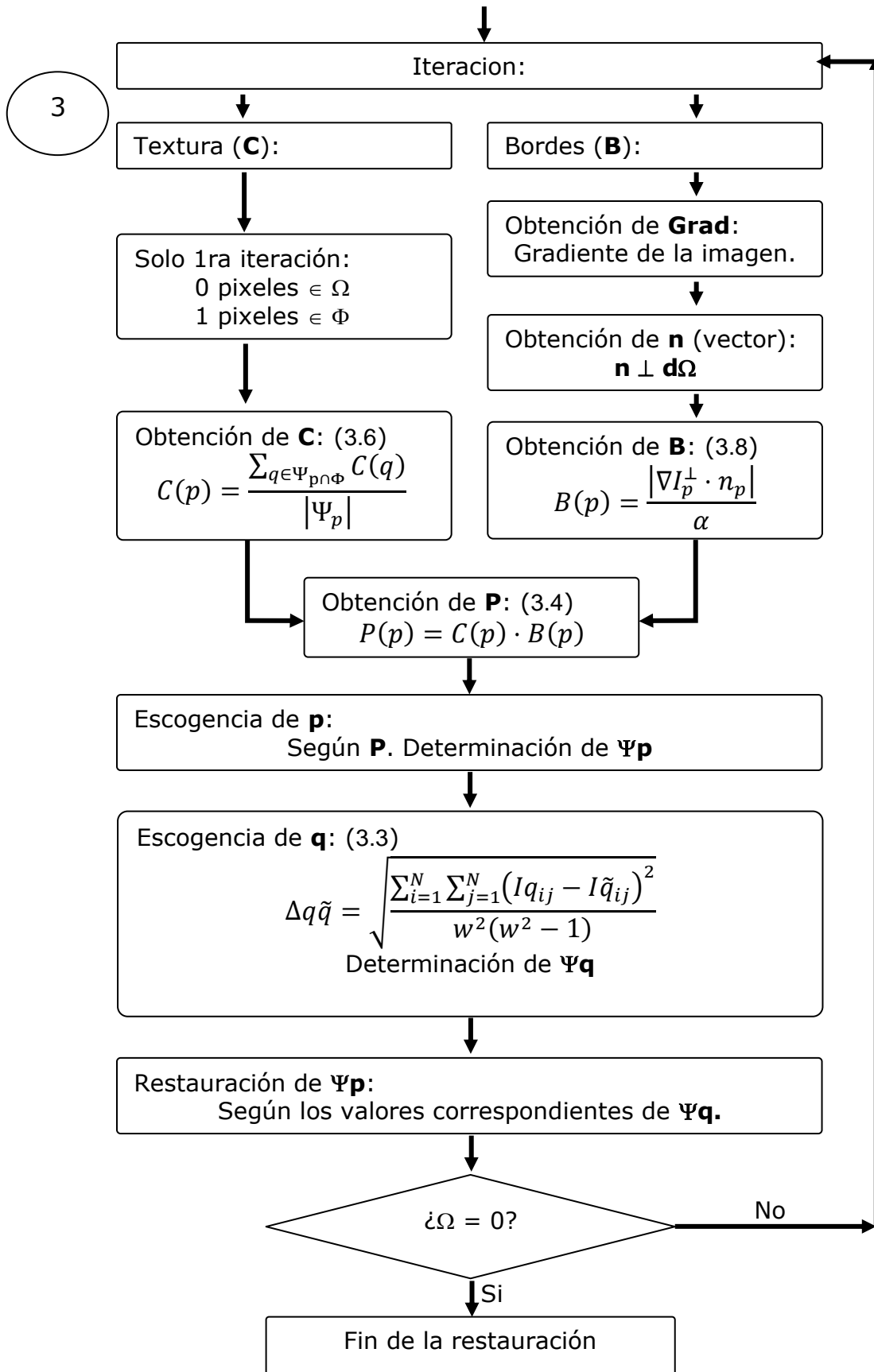
## Programa de cómputo:

1

Se cargan las variables de entrada:  
Imagen a restaurar.  
Imagen demarcada con la zona a restaurar  $\Omega$ .  
Código RGB del color que determina  $\Omega$ .

2

Se definen algunas matrices adicionales para facilitar las operaciones:  
Matriz **Ind**: Numera todos los pixeles de la imagen para facilitar su identificación.  
Matriz **Om**: Siempre contiene solo 1 y 0. Permite identificar a  $\Omega$  a medida que la imagen vaya cambiando.  
Vector **dOm**: Contiene los pixeles indexados según **Ind** de  $\delta\Omega$   
Matriz **Grad**: Guarda la información del gradiente.  
Matriz **n**: Guarda la información de los vectores perpendiculares a  $\delta\Omega$   
Matriz **C**: Matriz de confianza.  
Matriz **B**: Matriz de bordes.  
Matriz **C**: Matriz de probabilidad.  
Todas las matrices, menos **Ind** se actualizan en cada iteración.



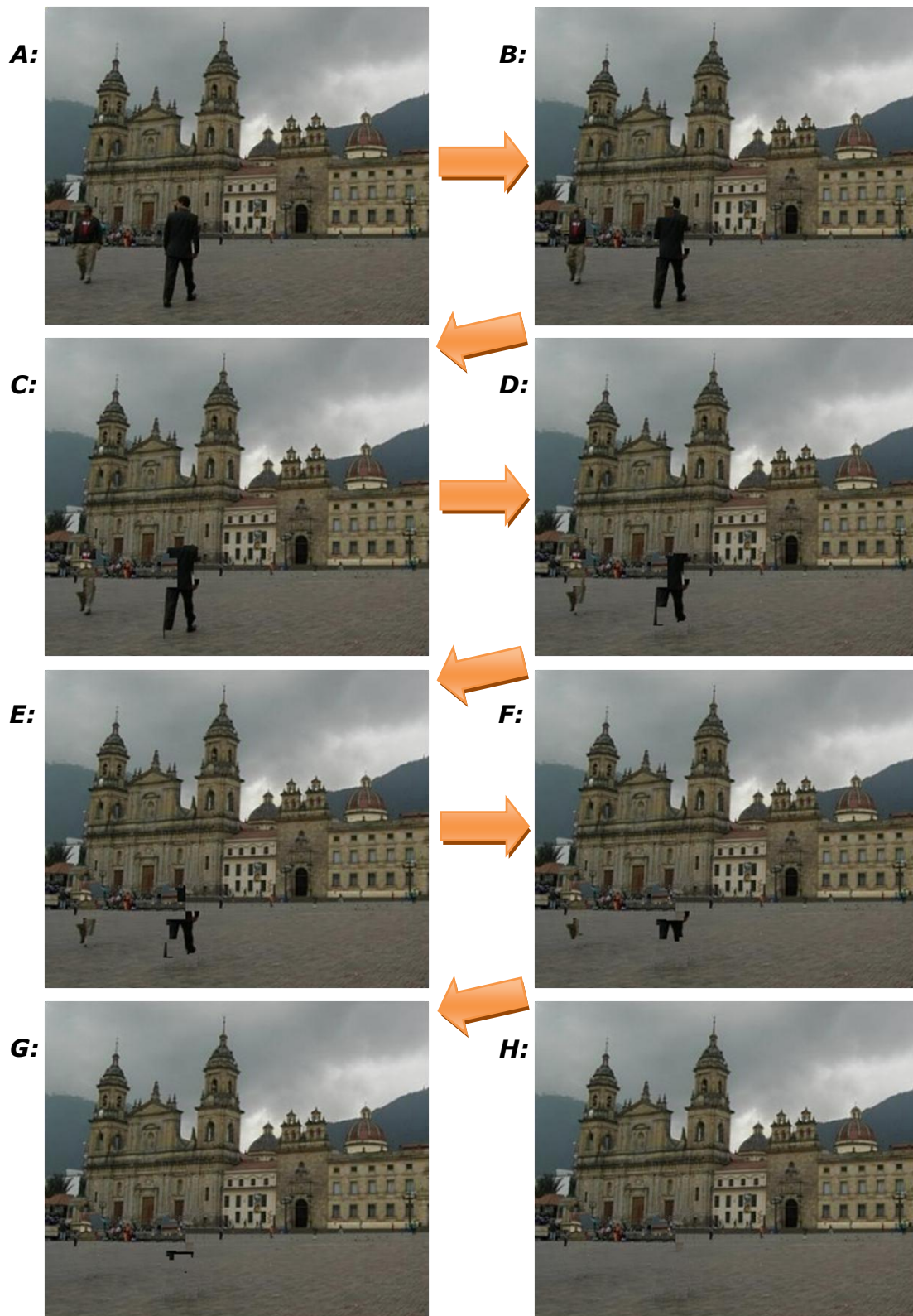
El siguiente paso es mostrar los resultados obtenidos en este trabajo.

### **3.2. Imágenes restauradas mediante el modelo propuesto.**

Ya se ha mostrado uno de los primeros resultados significativos de este trabajo, es decir la *Imagen 3.1-1* que corresponde a la plaza de Bolívar en Bogotá. En dicha imagen se ve que aunque es visualmente satisfactoria, no se reproduce la puerta de la catedral que está siendo tapada por la persona. Esto es de esperarse ya que el modelo permite hacer la restauración a partir de la información brindada por la frontera. Claramente no tiene que dar cuenta de objetos que están totalmente ocultos. Sin embargo, se encontró con satisfacción que variando algunos parámetros y repitiendo la restauración desde el comienzo, efectivamente se reprodujo la puerta. Este resultado se muestra a continuación de dos formas distintas. En primer lugar, en la *Imagen 3.2-1.*, se muestran varios pasos intermedios en el proceso de restauración. En segundo lugar, en la *Imagen 3.2-2* se muestra el resultado final en un tamaño mayor para que se pueda apreciar mejor por el lector.

¿Quiere decir esto que el modelo si da cuenta de objetos que estaban totalmente ocultos? No. Lo que se observa es que al variar parámetros, en un momento dado la otra puerta de la catedral fue tomada como la textura a propagar y simplemente se pegó en el sitio donde debe ir la puerta que está siendo tapada. En el Anexo 1 se encuentran los parámetros utilizados en esta restauración.





**Imagen 3.2-1.** Restauración. Fotografía Plaza de Bolívar 02 en Bogotá  
**A:** Iteración 0. **B:** Iteración 10. **C:** Iteración 20. **D:** Iteración 30. **E:** Iteración 40. **F:** Iteración 50. **G:** Iteración 60. **H:** Iteración 66.

**A:**



**B:**



**Imagen 3.2-2.** Restauración. Fotografía Plaza de Bolívar 02 en Bogotá  
Imagen a restaurar. **B:** Imagen restaurada.

**A:**

A continuación se muestran sistemáticamente los resultados obtenidos. Se hacen algunas observaciones particulares, aunque la idea es que en cada caso las imágenes sean lo suficientemente claras y el lector haga su propio análisis de resultados. Hay que tener en cuenta que en cada resultado, hay parámetros diferentes como el número de iteraciones del programa, el tamaño  $\square$  del parche y el tamaño de la imagen entre otros. Para no hacer tediosa la lectura del documento se deja dicha información en una sola parte. El Anexo 1.

La *Imagen 3.2-3* y la *Imagen 3.2-4* corresponden a la restauración de una fotografía de una de las entradas a la Universidad Nacional de Colombia. Sin duda la Universidad tiene otros espacios y circunstancias que dan una mejor imagen, pero se hizo el ejercicio de restaurar en la medida de lo posible, la que se muestra. Como se observa, la restauración se hizo en dos pasos. Primero se quitó el fuego y luego el carro de policía antimotines. En el Anexo 2 se muestra el resultado que se obtiene si se intenta quitar ambas cosas al tiempo. Como se observa, no es muy satisfactorio. Como se mencionó anteriormente la zona a restaurar es definida por el usuario y sin duda puede incidir en la calidad de la restauración. Este es un claro ejemplo de una restauración que es conveniente hacer en varios pasos. Algo similar sucedió con algunas de las imágenes que se muestran a continuación. Se restauraron por pasos para obtener mejores resultados.



**A**



**B**



**Imagen 3.2-3.** Restauración. Fotografía portería Universidad Nacional de Colombia. **A:** Imagen a restaurar. **B:** Imagen restaurada.





**A**

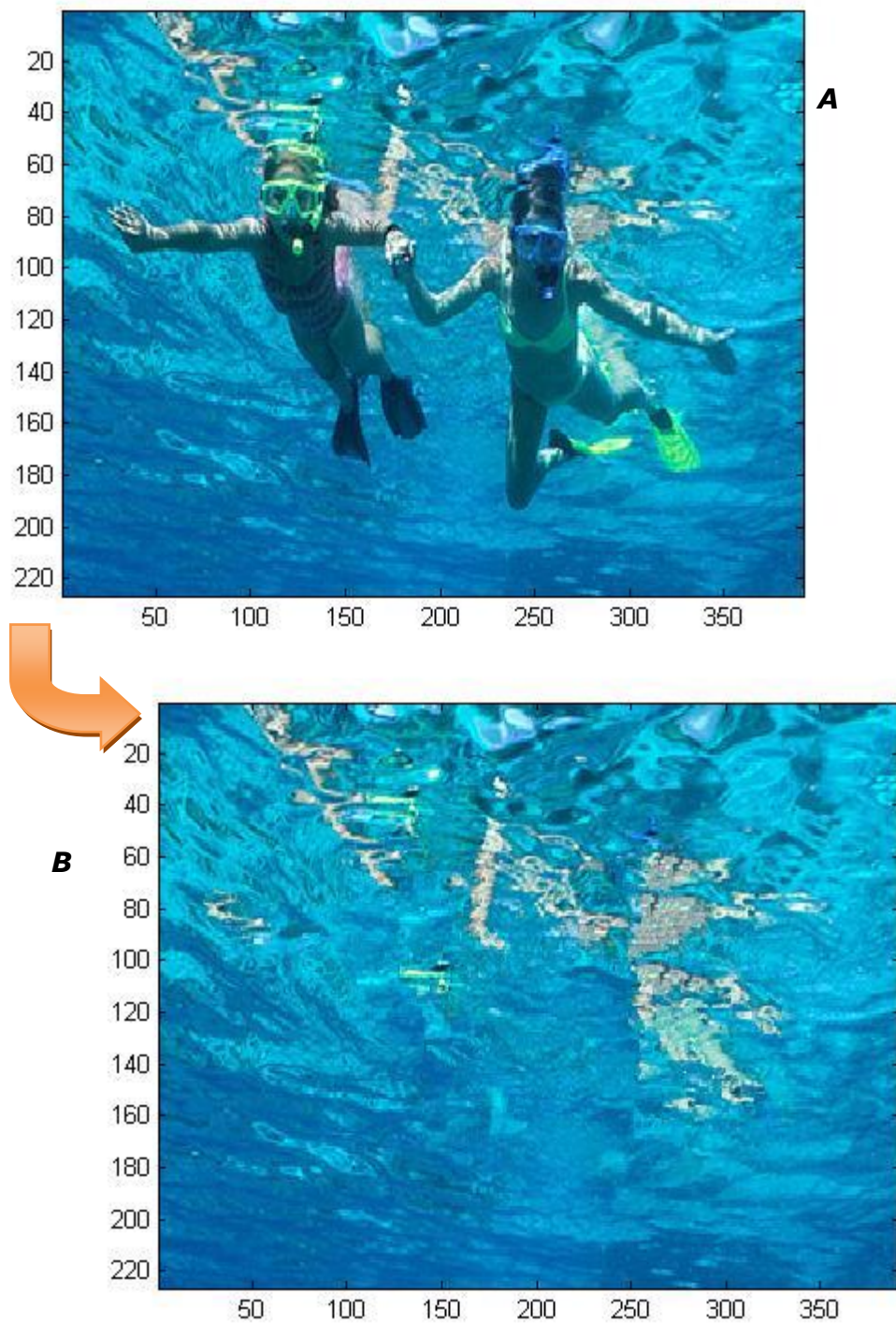


**B**



**Imagen 3.2-4.** Restauración. Fotografía portería Universidad Nacional de Colombia. **A:** Imagen a restaurar. **B:** Imagen restaurada.

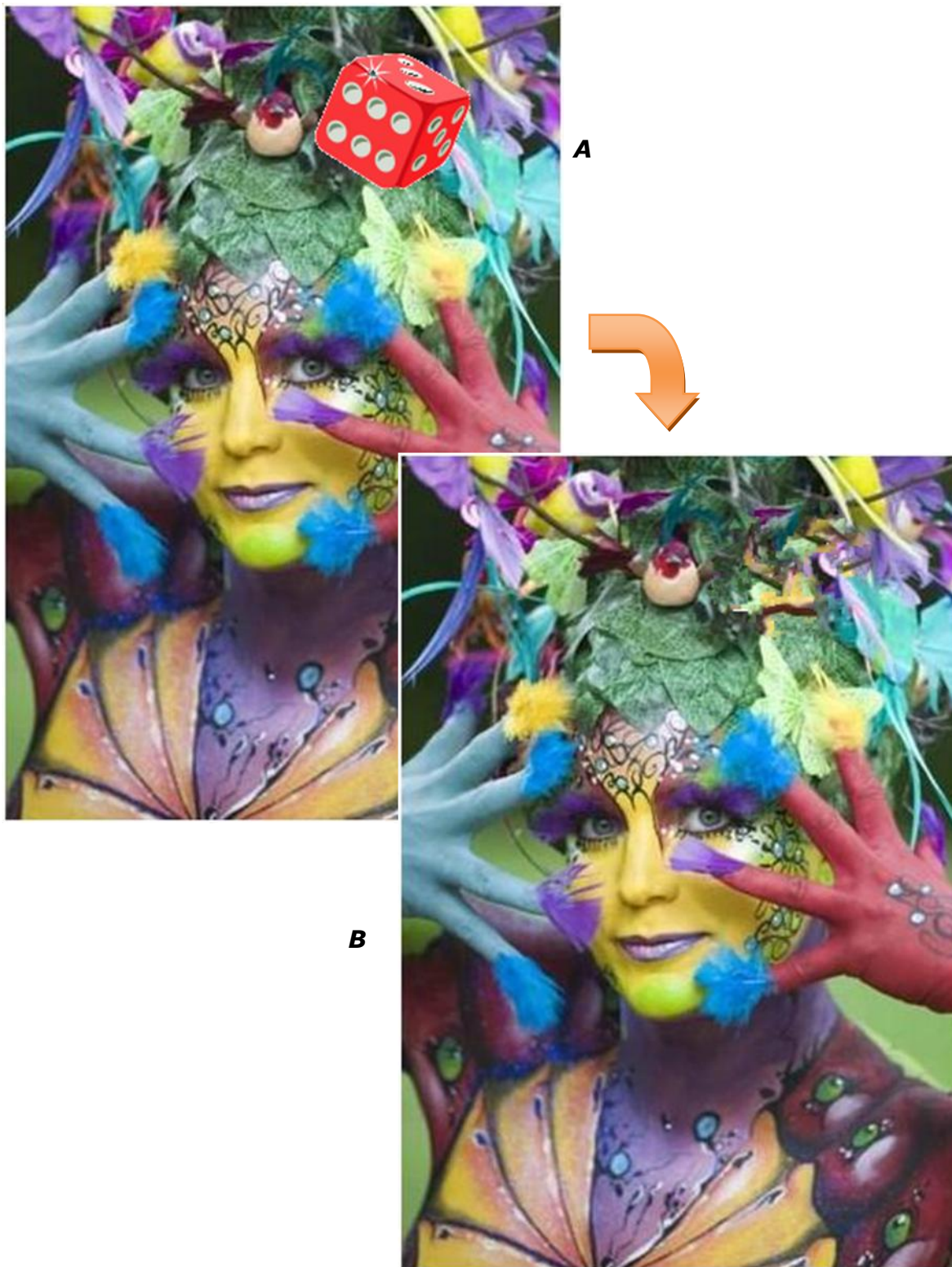
La *Imagen 3.2-5* corresponde a la restauración de una fotografía en el mar. En este caso, en forma similar a la de la *Imagen 3.2-1*, lo que se quiere es quitar las personas. El agua del mar puede ser considerada como una textura que es necesario propagar para restaurar la imagen. Se puede decir que el modelo arroja resultados satisfactorios desde el punto de vista de la restauración de textura.



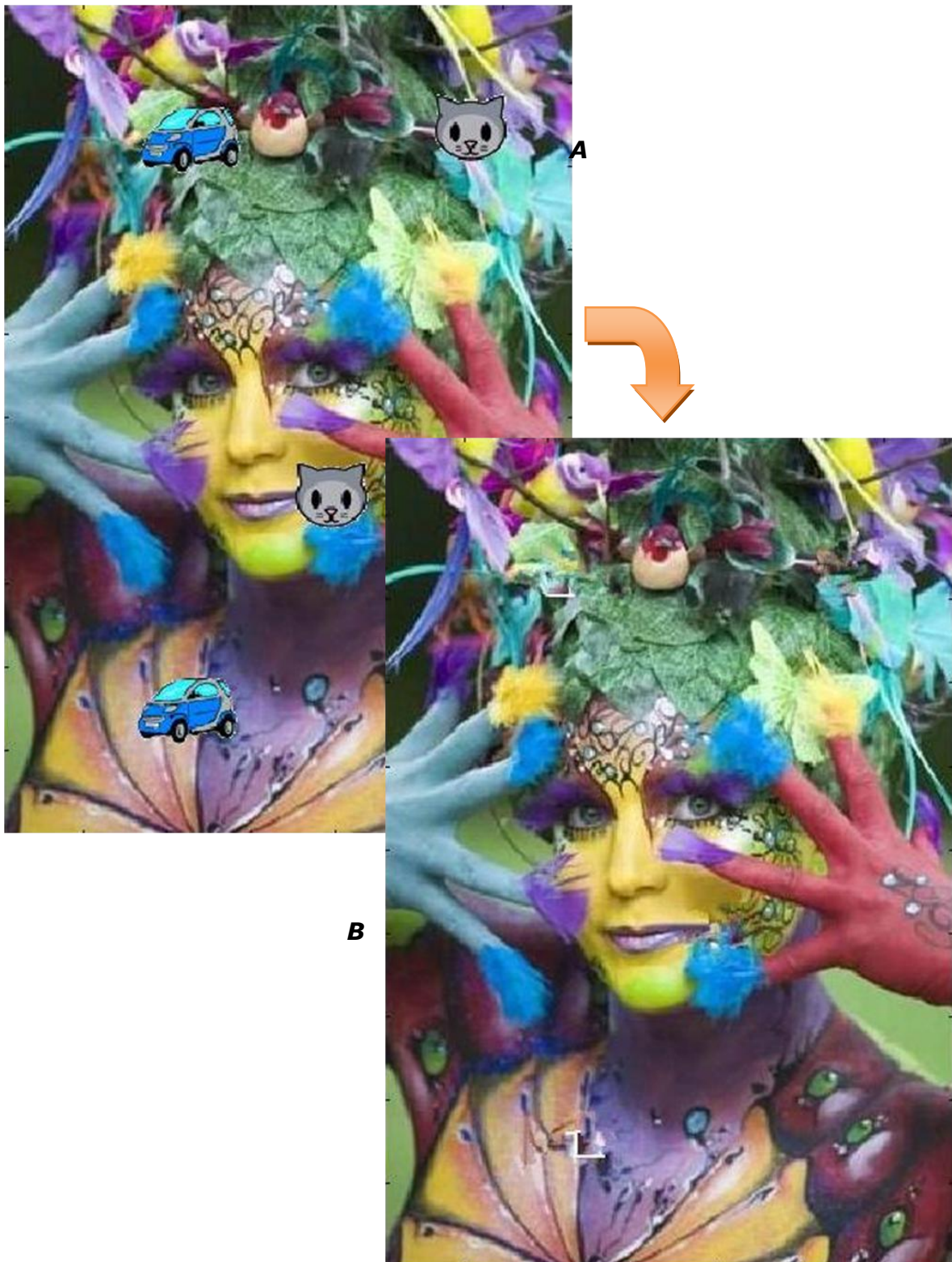
**Imagen 3.2-5.** Restauración. Fotografía en el mar. **A:** Imagen a restaurar. **B:** Imagen restaurada.

La *Imagen 3.2-6* y la *Imagen 3.2-7* corresponden a la restauración de una fotografía de una mujer con el cuerpo pintado. Esta técnica se conoce como Bodypainting. De nuevo la zona restaurada se puede considerar como una textura.





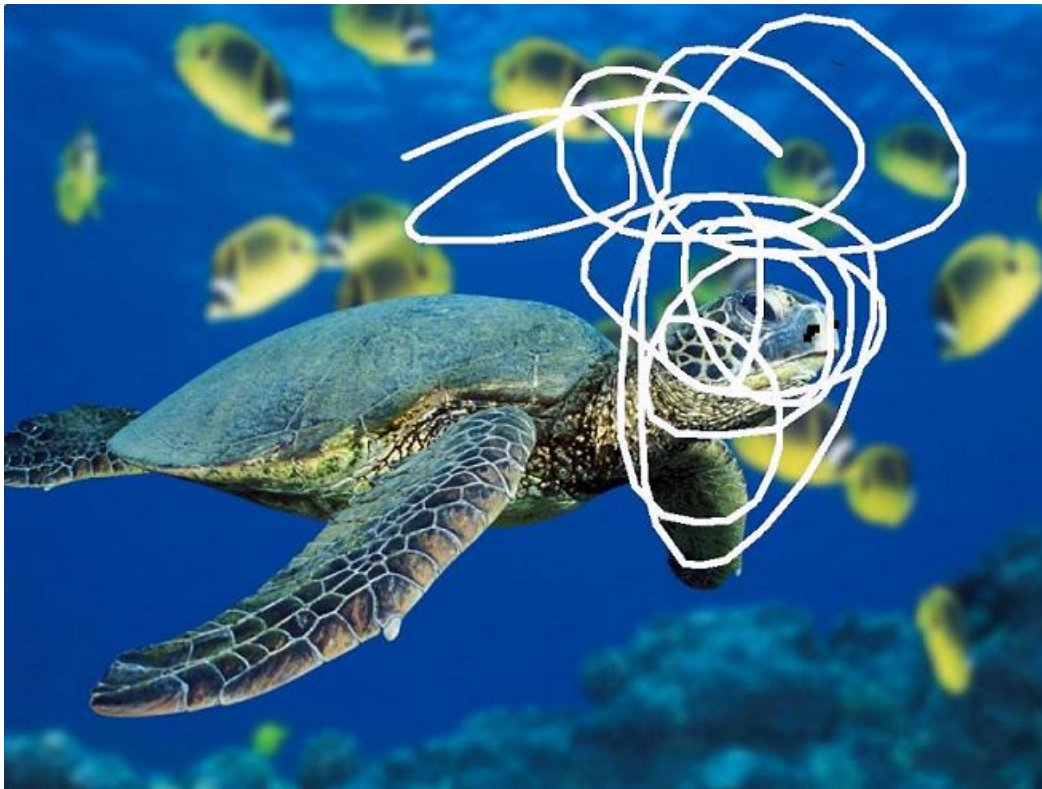
**Imagen 3.2-6.** Restauración. Mujer con BodyPainting. **A:** Imagen a restaurar. **B:** Imagen restaurada.



**Imagen 3.2-7.** Restauración. Mujer con BodyPainting. **A:** Imagen a restaurar. **B:** Imagen restaurada.

La *Imagen 3.2-8* es la restauración de una fotografía de una tortuga en el mar, la cual ha sido rayada.





**A**



**B**



**Imagen 3.2-8.** Restauración. Tortuga con rayón. **A:** Imagen a restaurar. **B:** Imagen restaurada.